Software Requirements Specification

For

Account Automation System

Prepared by

Samiul Basher (**170127**)

SM Faisal Nirjhor (**170145**)

Latifur Rahman Zihad (**180104**)

Md. Rahedul Islam (**180107**)

Md. Sahabuddin (**180108**)

Shariar Oni (**180127**)

Department of Computer Science & Engineering

Jashore University of Science and Technology

Date: 30/12/2021

I. Project Description

A. Project overview: Summary of the project

Without a budget management system, it is basically impossible to keep track of your money, provide accurate forecasts and reach your financial goals. The project is conceived based on the need to automate and integrate the financial budgeting with different administrative management systems of the university.

B. The purpose of the project

B1. Background of the project effort

Our main focus on a part of whole Account system which is budget segment.Main domain is from applicant to Vc sir's approval and provide relavant data to them to make easier than manual.

B2. Goals of the project

Without a budget management system, it is basically impossible to keep track of your money, provide accurate forecasts and reach your financial goals. The project is conceived based on the need to automate and integrate the financial budgeting with different administrative management systems of the university.

C. The Scope of the work

C1. The current situation

We have to build a good looking helpful Automated website which can solve the background issues of our account problem

C2. The context of the work

Such systems help to promote accountability and improve fiscal and financial discipline, by enabling thorough analysis of revenue and expenditure trends and components. As the level of university

administration information technology continuing to rise, to achieve information management of budgeting is essential.

C3. Domain analysis

Technical Literature:

Budget planning and administration are at the heart of good public financial management of any country or organization. Preparing executing, and tracking budgets means managing voluminous financial data, which can create a cumbersome and time-consuming budget process that hampers delivery of services. An efficient budgetary process largely depends on the existence of multi-functional information system for capturing budget data.

Existing applications:

This is an admin-based application software. Such kind of application are not available in Play store (for android devices). Therefore, such application is available for web application, but in the system, this is specially maken for university or any organization. There are some most popular budget management apps such as Scoro, Centage, Prophix, Float**,** Planguru**,** GIDE, Maxiplan, Tagetik**,** Adaptive Insights**,** Coupa**, Tidemark,** Neubrain, BillQuick, QuickBooks etc.

C4. Product features and Prioritization

Identifier	Priority	Requirement
REQ-1	5	The system will allow both the admin, user and recommending officer to create an account so that after verification each individual can log in to the systems.
REQ-2	5	The system will allow the user to fill up all required details as an online application
REQ-3	5	The system will notify corresponding officer if a new budget being registered. The system will also add as a pending budget on the officer’s desk.

REQ-4	5	The system will allow a corresponding officer to approve or reject or pending a budget based on his decision
REQ-5	3	The system will allow an officer to comment on his opinion about the budget he will be assigned to look after
REQ-6	4	This system will notify the next corresponding officer after an officer approve the budget. The system will also add as a pending budget on the new or responding officers' desk
REQ-7	5	This system will notify the applicant if is budget is approved or rejected by an officer
REQ-8	4	This system will notify the corresponding officer if a new budget comes into his authorization.
REQ-9	3	The system will show to which responding officer if the budget being pending in.
REQ-10	3	The system will send the budget towards the next department if all of the officers of the current department approved a budget
REQ-11	3	If a user is inactive for a while, the server will automatically log him out.

E. Stakeholders:

Admin:

VC sir:

Treasurer:

Recommending Officer:

Accounts Officer:

Deputy Director:

Director:

Teacher:

Staff:

University:

II. Requirements

A. Product use cases

A1. User classes and characteristics

User of this system should be able to apply for budget of their project in university, the respective sector officer.

User should able to the following function-

- Apply for budget
- Notify if the budget is rejected or keep pending

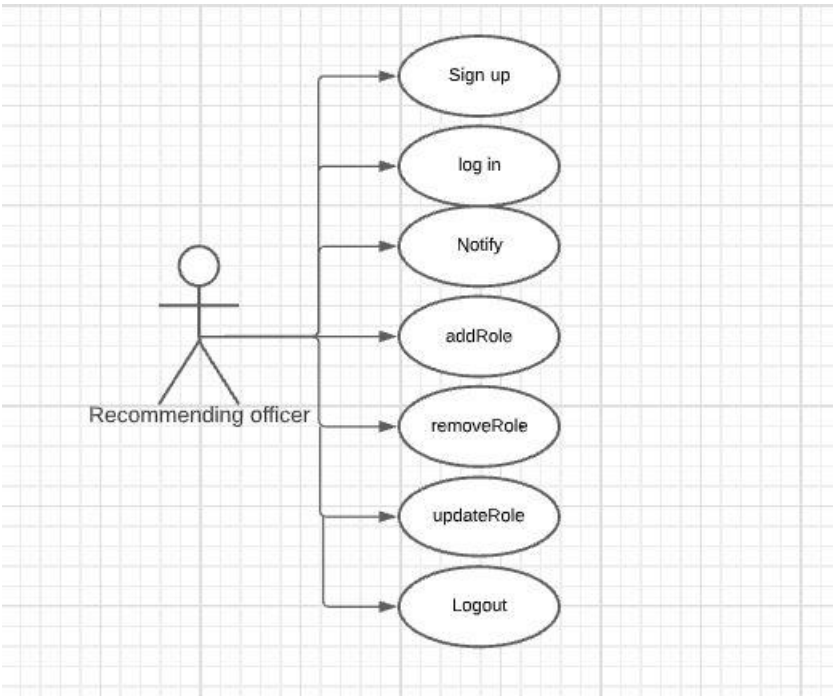
The recommending officer, VC sir, Treasurer, Recommending Officer, Accounts Officer, Deputy Director, Director should able to the following function

- Approve
- Reject
- Keep pending
- Comment
- See the list of pending
- Notify if the proposal come to him
- Can manually fill up the form

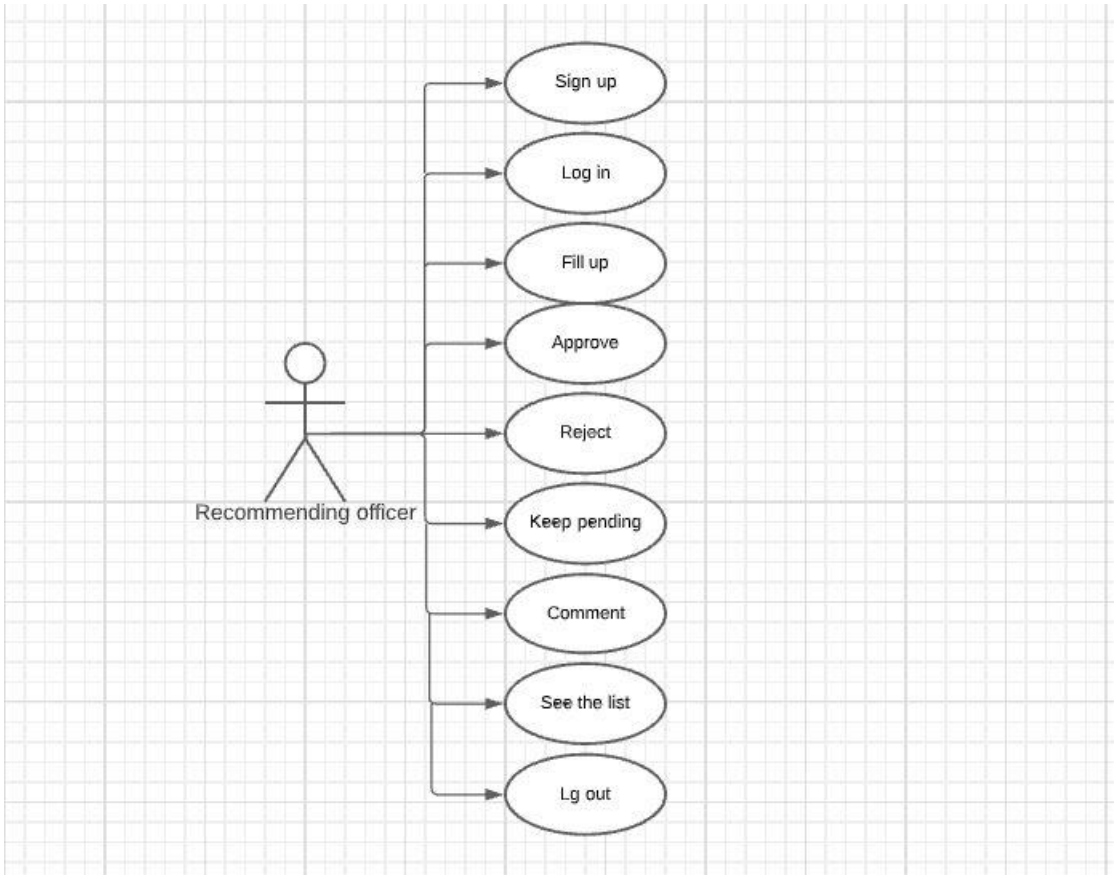
A2. User story

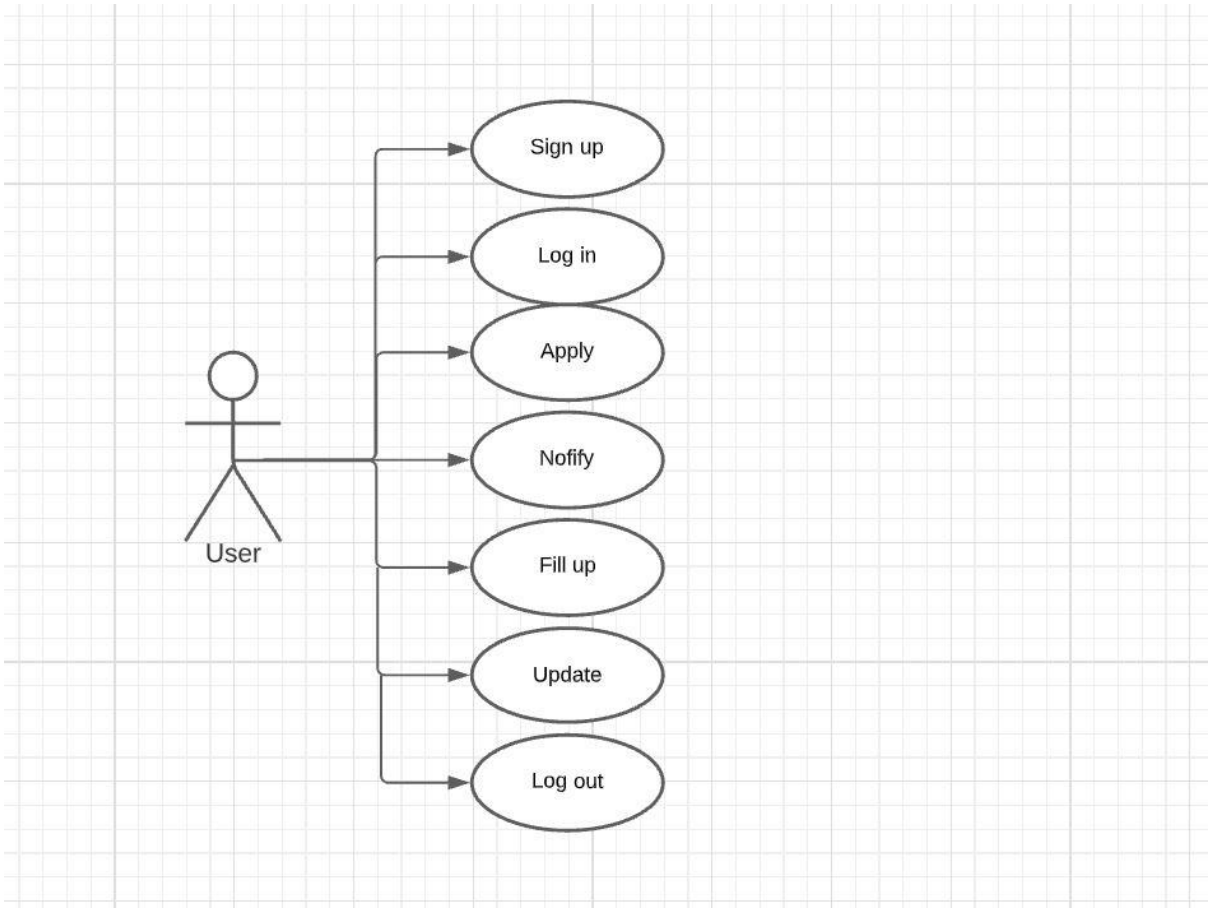
In the Accounts Automation System, the user he/she will register for account in order to apply for budget. The admin has whole access to the system to allow anything or manually fill up the form. The system will allow one profile to other in order to approve, reject or pending the proposal of the budget. The whole process will be notified to admin. It also stores previous history of the budget in each profile level.

A3. Use case diagram



Recommending officer





A4. Product use case list

Use case Name	Actor/Actors	Actor's Goal
Sign Up (UC-1)	Budget Holder, Accountant Officer	If they don't have a verified account, they have to sign up first with their real identity.
Log In (UC-2)	Budget Holder, Accountant Officer, VC,Treasurer	To enter their account for applying budget, seeing its condition (Budget Holder) and to see the budget application of the budget holder.
VerifyTheUser (UC-3)	Admin	When the user wants to sign up, verify them through email.
AddUser(UC-4)	Admin	When verification is completed, give them access to his account.
RemoveTheUser(UC-5)	Admin	When any user account validity expires, remove the account.
SubmitTheBudget(UC-6)	Budget Holder	If his budget list is complete, he will submit it for verification.

RecommendTheBudget(UC-7)	Recommendation Officer	To recommend the budget of the Budget Holder.
ViewTheBudget(UC-8)	Accountant Officer,VC	When the budget has appeared then view the information of the budget.
SwitchTheRole(UC-9)	BudgetHolder, Recommendation Officer	To switch his account.
RejectTheBudget(UC-10)	Accountant Officer, VC,Treasurer	To reject the budget
approveTheBudget(UC-11)	Accountant Officer, VC,Treasurer	To approve the budget.
LogOut(UC-12)	Budget Holder, Accountant Officer, VC,Treasurer	Log out from his account
Notify (UC-13)	Accountant Officer, VC,Treasurer	To notify the budget Holder
TimeOut(UC-14)	System	To log out automatically after some time.

A5. Individual product use cases

Use Case

Use Cases Description: In this section use case scenarios are described elaborately.

Use case name: Sign up

Primary Actor: User, recommending officer, admin.
Goal in Context: To register in the system.
Precondition:

- Accounts Automation system has been designed to add User, recommending officer, admin.
- It has an interface for registration.

Triggers: The User, recommending officer, admin have to register.
Scenario:

- Visit the register page.
- Input required information
- Check availability for username and check validity of password.
- Email sends to user e mail address.
- User confirm from his/her e-mail address.
- Confirmation message shown

Exception:

- Invalid Input.
- Not valid for Registration
- Authentication failed.

Priority: Must be implemented.
When Available: At the first time of registration.

Use case name: Log In
Primary Actor: User, recommending officer, admin.
Goal in context: When user want to enter the system.
Precondition: Must be registered.
Triggers: Need to log in the system for User, recommending officer, admin. **Scenario:**

- Visit the login page.
- Input Username and Password.
- Proceed the next activity.

Exception:

- Username/Email address invalid
- Wrong Password

Priority: Must be implemented.
When Available: First implemented and after log out for the system.

B. Operating environment

Operating environment for AAS is

- Operating system: WebOS
- Operating Database: mySQL
- Operating Platform: xampp

C. Design and implementation constraints

AAS will be a web-based application system. The system will be developed using web technology and web server.

A person with no technical knowledge will also understand the system if he knows the university's budget system properly.

D. Assumptions and dependencies

- Each user must have an account
- Users must log in the system to access on any record
- Admin have ability to add or remove anyone or any role.

E. Functional requirements

. **Requirements**:

Normal Requirement: Normal requirement consists of objectives, goals, minimal function and performance that are stated during the meeting with customer. The normal requirements of our system are:

- The system will allow both the Admin & User to create an account so that after verification each individual can login into the system.
- The system will allow specific officer (in case a budget request comes in offline) to fill up all required details as

an online form to add a request of a new budget. The form will have to contain option to attach files.

- The system will notify the corresponding admin if a budget being registered. The system will also add as a pending budget on the corresponding admin's desk.
- The system will allow the main admin or the in charge of a department to approve or reject or keep pending based on his decision.
- The system will send an email to the User if the budget is rejected or get pending. Or the admin will send an email manually if the case is sensitive.
- The system will allow the officer to comment his opinion about the budget he will be the assigned to look after.
- The system will notify if the next officer if an admin approves the budget. The system will also add as a pending budget on the corresponding admin's desk.
- The system will notify the corresponding admin if a new budget comes into his authorization.
- The system is worked by serially. That's why the system will send the budget towards the next department after approving this.

Expected Requirement: Expected requirement consists of implicit requirements and may be so fundamental that the customer does not explicitly state them. Their absence will be a cause of dissatisfaction. This is a part of System requirements. The expected requirements of our system are:

- Accessible via internet
- Verify email
- Manage database for Admin & User
- Manage database for budget

- Dashboard for Admin & User

Exciting Requirement: These requirements are for features that go beyond the customer's expectation and prove to be very satisfying when present. The exciting requirement of our system is:

- The system will allow to track the budget by the user which he submitted.

F. Data requirements

G. External interface requirements

G1. User interfaces

The user interface is designed in web OS. The developer will have to study the designing of the product.

G2. Hardware interfaces

- **Laptop/PC/Desktop**
- **Android/IOS/Tablet**

G3. Software interfaces

Software used	work
Operating system	We have chosen web OS.
Database	To store we have chosen mySQL

XAMPP	**To implementing we have chosen XAMPP**

G4. Communication interfaces

The system will support all digital device which support web browser.

H. Non- Functional requirements

H1. Performance requirements:

The application should have high performance and low failure rates. The application should have high security and encrypted so that no hacker can access or hamper the budget

H2. Dependability requirements

The application should have 95% of access in time to time. The rest 5% when this system has too much load that will be a rare case. The failure rate only 5%.

H3. Maintainability and supportability requirements

If the automated email services become unavailable, they can be under maintenance for approximately three hours.

H4. Security requirements

All the user has a unique login. And system encryption is high enough to secure the data from hackers.

H5. Usability and Humanity requirements

The system interface has to be user-friendly and easy to use.

H6. Look and Feel requirements

The system should have a front-end design that looks like administrative and feel like a professional administrative officer.

H7. Operational and Environmental requirements

The system should have cross platform benefits. It must be web-based platform.

H8. Cultural and Political requirements

The system does not disrespect any cultural or constitutional things of our nation.

H9. Legal requirements

The system must follow all rules of its types system and also legalized by authority.

V . Test Plan:

A.Scope of Testing

This section is important to include because it lists the overall areas included and excluded from testing. It provides an insight into the approved scope of work for the QA team and works as an excellent reference for reporting.

1.**Features to be tested :**

a) Create a new account.

b) Delete an account.

c) Add budget information.

d) Calculate budget cost automatically.

e) View budget information.

f) Reject /approve budget .

g) Recommend the budget holder.

h) Seeing the budget position.

i) Authenticate user

j) Auto logout

k) Server response time.

l) SignIn.

2.**Features not to be tested :**

a. Simplicity.

b) Useability.

c) Efficiency.

d) Durability.

B.Pass/Fail Criteria :

Features	Testing Attributes	Pass/Fail
LogIn	Email=shariaroni@gmail.com Password=12345	Pass
LogIn	Email= rahedul0176@gmail.com	Fail(invalid pass)

	Password= Null	
SignUp	Name=X, email=x@ymail.com, Mobile=017688499, Designation=Budget holder, Password = 12345	Pass
SignUp	Name=X, email=x@ymail.com, Mobile=017688499, Designation=Budget holder, Password = 'Null'	Fail
BudgetApply	Criteria = Prepaid Financial apply Elements = x,y,z Price =10000, Pre need =5000, Recommondent = ADept CharmanSir	Pass
Recommended	User ID = 01, BudgetList = userId.budget Recommended =Yes	Pass
Recommended	User ID = 01, BudgetList = userId.budget Recommended =No	Fail
Authenticate	Email=valid, PasswordOfVarifcation =Valid,	Pass
Authenticate	Email=valid, PasswordOfVarifcation =Wrong,	Fail
Calculate cost	Elements=1,price=100,unit=3, Elements=2,Price=200,unit=2, Total cost=700	Pass

C. Testing Approach

Approach 1. First we have to have a stable network connection . Then We have to decide all of the information for the project.

Approach 2. Then we started to test step by step for every test case.We test every page step by step.

Approach 3. When single steps have been done then we started to test a few test cases jointly. Then if we get any bug we send it to the other team members and fix it.

Approach 4. Testing every test case with different values such as null,error,int,string etc. answers see what will happen and how it will notify the user .

Approach 5. We tried to test basic things fast then margely tested all the features.

Approach 6. We test about 3 or 4 rounds for every test case.

Approach 7.We also test defect test ,stress test,performance test and so on.

List of tools : GitHub,Visual Studio,PHP jam server etc.

D.Requirement Traceability :

REQ'S	PW	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11
REQ-1	5	x	x	x	x							
REQ-2	5			x	x							
REQ-3	5								x			
REQ-4	5								x			
REQ-5	3											x
REQ-6	4					x						x
REQ-7	5											
REQ-8	4						x			x		
REQ-9	3	x						x				
REQ-10	3			x							x	
REQ-11	3											
MaxPW		5	5	5	5	4	4	3	5	3	3	4
Total		8	5	13	10	4	4	3	10	3	3	7

E : Suspension and Resumption:

According to this we need to specify the criteria that is used to suspend all or a part of the testing activity on the items of testing that are associated with the plan. Also specify the testing activities that must begin on resumption.

It is important to understand that if a defect is detected at a point after which the testing shall resume proves to be of no use, then applying

the resources on testing will be futile. One needs to specify the reason for stopping the test activities and define the acceptable level of defects that allows the testing process to surpass those defects. Few defects may arise on resuming the test activity. Those are the hidden defects that were somewhere ignored earlier.

F . Testing Materials:

1. Computer .
2. Browser which supports our application..
3. TestProject(testing software).
4. Software must be useful for our web application.
5. Operating system Web OS

G. Test cases:

Test-Case Identifier :	TC-1(login)
Use Case tested :	UC-2,main success scenario,UC-3
Pass/Fail Criteria :	The user pass if he had given valid information and if he tried less than three times
Input Data :	Email,password
Test procedure	**Expected Result**
1.Fast go to the web page and click the login button. 2.Give valid information with user email and password.	If he enter valid password then open his account interface and can apply for his budget (budget holder) and approve the budget(Accountant officer).

Test-Case Identifier : **Use Case tested :** **Pass/Fail Criteria :** **Input Data :**	TC-2(SignUp) UC-1,main success scenario,UC-3,UC-4 The user pass if he had given valid information and if he tried less than three times Email,password,mobile number,Name
Test procedure	**Expected Result**
1.If he don't have any account he will click the sign up button 2.Give valid information with user email and password then if once verification is completed his sign up has been completed.	If he enter valid password and all other information then open his account interface and can apply for his budget (budget holder) and approve the budget(Accountant officer) and they are able to login into his account.

Test-Case Identifier :	TC-3(ApplyBudget)
Use Case tested :	UC-6,main success scenario,UC-7
Pass/Fail Criteria :	The user passes if he had given valid information and if his given information is totally completed.
Input Data :	Budget category,Cost,Elements,etc
Test procedure	**Expected Result**
1.He will click on the application button then go to apply for the budget section. 2.Give valid information of his applied budget then it is submitted for approval.	If he gives the valid information the budget application will be sent to the accountant officer.

Test-Case Identifier :	TC-4(Recommended)
Use Case tested :	UC-7,,UC-10,UC-11
Pass/Fail Criteria :	

Input Data :	The user passes if he had given valid information and if the recommondened officer approve but the next officer is a failure criteria. Budget information,approved
Test procedure	**Expected Result**
1.He will click on the approve application button then go to the budget section officer for approval. 2.If it approves then send it to the accountant officer.	If he gives the valid information the budget application will be sent to the accountant officer.

Test-Case Identifier :	TC-5(Authenticate)
Use Case tested :	UC-2,UC-3,UC-9
Pass/Fail Criteria :	The user passes if he has given valid information and matches with his password he will login into his account .

Input Data :	Password,email
Test procedure	**Expected Result**
1.When he gives valid information it will compare with his database information and if it is right log into the information. 2.If it approves then he will be able to enter his room.	If he gives the valid information then he will be able to log into the account.

Test-Case Identifier :	TC-6(calculate Cost)
Use Case tested :	UC-11
Pass/Fail Criteria :	The user passes if his given total cost is correct.
Input Data :	Cost,Elements unit etc
Test procedure	**Expected Result**
1.When a budget user applies his cost.He will input the data and give the total value of cost .	If he gives the valid information then it calculates the budget holder total cost.

H . Testing strategy :

Unit test: Unit testing is a type of software testing where individual units or components of a software are tested.In our testing part firstly we test our software each and every part individually.We will use the unit test in every part.

Integration Test: Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.Firstly in our testing we use unit test then we test it in integration test.We use a few tests of the attributes and use the integration test.

Performance test : Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size.

Examine speed:Our examine speed is too fast .and we can load every page fastly.

Reliability : Our reliability of our web application is very good.

Application size:Our application size does not matter for the user.He only needs a browser and he can use it easily with the help of a server.

Stress Test: Stress Testing is a type of software testing that verifies stability & reliability of software application. The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software doesn't crash under crunch situations.In our application we test the durability of its server capacity etc.It helps us to enhance the performance of the project.

Project Issue

A project issue is a problem that has been encountered in executing project activities. This problem impairs a project’s ability to successfully
complete.

A. Task

Server Issues

When using external Web hosting, the hosting company's servers

may experience downtime. When the servers are down, our site is down, and when our site is down which translates into time loss. To reduce the risk of server issues, choose a hosting company with backup servers, and make sure it guarantees at least 99.5 percent up-time for its regular servers.

Bandwidth Issues

Most hosting companies place a limit on the amount of bandwidth for each month. But for every month we don't need same bandwidth. If we anticipate that our site may eventually exceed the bandwidth allotment, choose a host that offers the option to upgrade to a dedicated server. A dedicated server is our own server with unlimited bandwidth. Otherwise, we have to go through the hassle of moving your entire website to another host.

Dynamic IP Address Issues

The domain name for our site is assigned a "dynamic" Internet protocol address, or IP. An IP address is a series of numbers associated with your domain name. A dynamic IP address is shared with numerous other website owners, which decreases

the stability of our website. Dynamic IP addresses are always changing, which also decreases the stability of our site.

Static IP Address

To prevent a dynamic IP address problem, consider purchasing a "static" IP address, which is our own IP address -- it never changes, and is not shared. This results in more site stability. To purchase a static IP address, contact our Web hosting provider or Internet service provider. If purchasing through our ISP, request that the ISP associate the static IP address with your e-businesses domain name.

Security Issues

In our system security is absolute priority.Hackers can break into our website's database to steal confidential and proprietary information, along with user data. To prevent hacks and to increase user confidence, we have to secure our website by purchasing a secure socket layer, or SSL, certificate. Most Web hosting companies offer SSL certificates for an additional fee.

B. Risk
code issues
One significant risk involved with software development is poor quality
code. Projects may contain poor quality code because of rushed work

and many other factors. Issues with code may include bugs, logical errors and more. You can mitigate risks related to code quality by:
Testing code frequently
Resolving bugs and logical errors when they're found
Creating coding standards for software developers
Using coding best practices

Online Security Breach
our site is vulnerable to online security breaches and cyber-attacks. Some of these online security risks can include phishing, website hacking, and unprotected web services.
There are many hackers who can breach the network of a company and
access sensitive information.
Therefore, it is necessary that your e-commerce website security is very
strong.
Aggressive deadlines
Sometimes, software development projects have tight deadlines. In some cases, software development teams may be unable to meet these deadlines. We can mitigate this risk by creating a thorough project plan that allows you to set realistic deadlines.

Low productivity
Productivity issues can also be risks in software development. Sometimes, software development teams may struggle with productivity, which can happen because of delays, employee

burnout and many other factors. You can increase your team's productivity by:
Creating a well-paced project plan to lower stress and avoid

burnout
Communicating effectively about project details and problems
Find a great leader who can motivate and manage the team

Credit Card Scams
Suspicious transactions and stolen credit card information are common
risks of online payment.
Our online security should be strong and tight enough to catch a doubtful transaction.

Weak Authentication Methods
If we have weak and very basic authentication methods, then you are prone to more cyber-attacks.
If we are authenticating a user by ID and password only then there are chances that this information can be stolen.
we need strong authentication methods for your online security that can resist attacks.

Scope creep
Changing project scopes can also cause risks in software development.
Scope creep refers to a project's scope morphing into something

completely different than it was initially. Scope creep can cause risks when it causes software developments to miss project deadlines and extend project timeframes. We can monitor scope creep by separating our project into manageable segments or iterations and frequently reviewing the scope.

C. Cost
We can't decide on a cost of the design process!
Even for the most experienced website designers, telling the cost of a project can become extremely difficult at times. Honestly, even you would find it impossible to exactly outline what your e-commerce site needs and what it does not at this moment. Getting a quote or

estimation before beginning a website design project is a cost guide at best.

One way you can overcome the uncertainty of cost is by laying down a
budget upfront. You simply need to prioritize desired features and functions as per cost to adhere to a particularly strict budget.

D. Ideas for solution
Solid requirements – clear, complete, detailed, cohesive, attainable, testable requirements that are agreed to by all players. In 'agile'-

type environments, continuous close coordination with customers/end-users is necessary to ensure that changing/emerging requirements are understood.
Realistic schedules – allow adequate time for planning, design, testing, bug fixing, re-testing, changes, and documentation; personnel should be able to complete the project without burning out.

Adequate testing – start testing early on, re-test after fixes or changes, plan for adequate time for testing and bug-fixing. 'Early' testing could include static code analysis/testing, test-first development, unit testing by developers, built-in testing and diagnostic capabilities, automated post-build testing, etc.
Stick to initial requirements where feasible – be prepared to defend against excessive changes and additions once development has begun, and be prepared to explain consequences. If changes are necessary, they should be adequately reflected in related schedule changes. If possible, work closely with customers/end-users to manage expectations. In 'agile'-type environments, initial requirements may be expected to change significantly, requiring that true agile processes be in place and followed.

Communication – require walkthroughs and inspections when appropriate; make extensive use of group communication tools – groupware, bug-tracking tools and change management tools,

intranet capabilities, etc. Ensure that information/documentation is available and up-to-date – preferably electronic, not paper; promote

teamwork and cooperation; use prototypes and/or continuous communication with end-users if possible to clarify expectations.

E: Project Retrospective

Before the Retrospective

Don't rush into the nearest meeting room the day after the launch and rattle through a cobbled together agenda just to tick off: "project retrospective". This is an opportunity to gain useful insights and learnings so a little planning will make all the difference.

What, When and Who?

Retrospectives happen at the end of a website project, after we've delivered the website to the client or our team. They don't need to happen immediately after, but ideally, we should take place within a fortnight of finishing the project, so everyone still has the work fresh in
their memory. Pick a time when everyone can participate, including the
most relevant people in our team, our client's team and all key decision
makers and stakeholders.

Send Out Prep

A few days before the retrospective, put together an agenda for the meeting and circulate it amongst all the people who were involved in the project. This will give them the chance to reflect on the project and
gather their thoughts in preparation for the retrospective. If we need anyone to bring anything in particular, now is the time to make that happen. It may be information on the budget, timeline, client feedback,
or anything else that will be key to one or more of the agenda items.

During the Retrospective

However, we decide to deliver our retrospective, there are some key

agenda items that are recommended as a minimum for the discussion: Discuss what went well, giving praise to those responsible if it feels fair
to do so. Discuss what didn't go well. What didn't go to plan and what
was changed? Be careful not to blame any specific individuals, but rather give constructive feedback if appropriate. Identify what we can learn from both the highs and lows so that we can change or improve things on our next project. Think if anyone else could gain value from hearing about your experiences, such as colleagues who weren't on the
project or workers in the industry who may enjoy reading your findings
as a blog post.

F: Open Issue
Limited User: In our software a small number of user can use it.
No Instant Verification: When a user will fill up sign up form he or she will wait until admin approved it.

Not for Any People: Limited people can create and use this system.
Not All Time Approved: Application will approved only office time.

G: FUTURE ENHANCEMENT
Accounting Automation System would help each user to move analog to digital system. User can apply for budget through internet and he or she can also check the stage of budget any time. By doing this this software file pending will be totally removed. User can feel a hassle free
budget experience. User can keeps budget history.

VIII. Learning during the project training.

Learning during the project training.

- **Learn the basics of programming**

Our Software Engineering Degree Apprenticeship programme assumes you have no previous knowledge of computing and teaches apprentices the foundations. If you're motivated and dedicated, you will learn the foundations of programming.

- **Understand the Software Lifecycle**

The course equips apprentices with the knowledge and experience to excel as software engineering practitioners, applying skills in disciplines such as Human-Computer-Interaction, User-Centred Design, Project Management and Software Engineering processes, and Advanced Programming in PHP. Become an expert in the different stages of the software lifecycle and you will be able to connect with all members of your team.

- **Become a better leader**

Everyone that takes the Software Engineering programme not only learns frameworks for leadership and change management, but you will put them into practice during the 6 month capstone project. Supported by an academic supervisor, this six-month, work-based project pulls together the skills and knowledge developed over the course of the apprenticeship, and provides a platform for apprentices to bring direct benefits to your organization.

- **Get an understanding of emerging tech**

By understanding the emerging trends of emerging tech applications, we can become a driver of change in your business. Taking part in the Software Engineering Degree Apprenticeship and understanding emerging tech, you will be able to influence your team's decision making and inspire them to push the boundaries of thinking when it comes to development.

- **Understand ethical and security concerns**

The security of your business is naturally a top priority as a project leader. During Covid-19, 91% of businesses reported an increase in cyber-attacks with employees working from home highlighting the need for greater focus on security. Apprentices will learn security principles relevant to software projects, from requirements and design, through to development and testing. This allows you to ensure that cyber security is at the heart of your software development projects from the beginning.

- **Learn how to be more innovative as a business**

With greater IT knowledge, you will be able to put IT at the centre of projects and manage a team in a way you have never been able to do before. The Software Engineering programme will allow you to create innovative website changes, app updates and smarter marketing campaigns that will set you apart in your industry.On top of this, the Software Engineering Degree Apprenticeship will also enable you to network, share ideas, and best practice with peers from a broad range of organisations in the North East. Apprentices can foster connections with both industry and leading academics in computer science to further innovation

Gather Experience to develop a software :

This describes an experience in delivering software development project courses . The objective is to propose a learning process model in a Software Engineering course. This model can be an alternative in the learning process that has improved our knowledge and skills in software development practices. According to the study program curriculum, an ability in the development of small to medium scale software is one of several learning outcomes that must be achieved by our graduate. To achieve this learning outcome, we give practical experience in applying one method to develop a medium scale

software through this development . This course is conducted in the form of a ‘real’ software development project.

Collaborate With team Members :

We are allocated into several groups to give an opportunity to work in a team. One medium software development project is assigned to each group. Since each project is part of a larger project, completion of this medium scale software development project will produce a large scale software system. Using the iterative and incremental approach known as the Unified Process, each group conducted a full software development life cycle: defining a software requirement specification, requirement analysis, design modeling, coding, and testing.
Evaluation of this course in several semesters showed that the project-based courses can improve the students' understanding about software engineering. The Unified Process is considered to be an appropriate method for this software development project. Despite the several obstacles that were still encountered, this course model can still be improved to achieve the ultimate goal of this course.Collaborate with team members and do the work.

IX. Glossary