###### 

**PUCIT**

Punjab University College of Information Technology



**First Deliverable**

**Version 1.0**



**TABLE OF CONTENTS**

[**FIRST DELIVERABLE GUIDE 3**](#_46r0co2)

[1 Introduction 3](#_2lwamvv)

[1.1 Project/Product Feasibility Report 3](#_111kx3o)

[*1.1.1 Technical Feasibility 4*](#_3l18frh)

[*1.1.2 Operational Feasibility 4*](#_206ipza)

[*1.1.3 Economic Feasibility 4*](#_4k668n3)

[*1.1.4 Schedule Feasibility 4*](#_2zbgiuw)

[*1.1.5 Specification Feasibility 5*](#_1egqt2p)

[*1.1.6 Information Feasibility 5*](#_3ygebqi)

[*1.1.7 Motivational Feasibility 5*](#_2dlolyb)

[*1.1.8 Legal & Ethical Feasibility 5*](#_sqyw64)

[1.2 Project/Product Scope 5](#_3cqmetx)

[1.3 Project/Product Costing 5](#_1rvwp1q)

[*1.3.1 Project Cost Estimation By Function Point Analysis 6*](#_4bvk7pj)

[*1.3.2 Project Cost Estimation by using COCOMO’81 (Constructive Cost Model) 8*](#_2r0uhxc)

[1.4 CPM - Critical Path Method 9](#_1664s55)

[1.5 Gantt chart 12](#_3q5sasy)

[1.6 Introduction to Team member and their skill set 13](#_25b2l0r)

[1.7 Tools and Technology with reasoning 13](#_kgcv8k)

[1.8 Vision Document 13](#_34g0dwd)

[1.9 Risk List 14](#_1jlao46)

[1 Introduction 15](#_43ky6rz)

[*1.1 Systems Specifications 16*](#_2iq8gzs)

[*1.2 Identifying External Entities 17*](#_xvir7l)

[*1.3 Context Level Data Flow Diagram 17*](#_3hv69ve)

[*1.4 Capture "shall" Statements 17*](#_1x0gk37)

[*1.5 Allocate Requirements 18*](#_4h042r0)

[*1.6 Prioritize Requirements 18*](#_2w5ecyt)

[*1.7 Requirements Trace-ability Matrix 18*](#_1baon6m)

# 

## ***1 Introduction***

First part of this deliverable is all about planning and scheduling of projects. This deliverable must contain following artifacts:

1. Project Feasibility
2. Project Scope
3. Project Costing
4. Critical Path Method Analysis (CPM Analysis)
5. Gantt Chart
6. Introduction to team members
7. Tools and Technologies
8. Vision Document
9. Risk List

## ***1.1 Project/Product Feasibility Report***

When a project is started the first matter to establish is to assess the feasibility of a project or product. Feasibility means the extent to which appropriate data and information are readily available or can be obtained with available resources such as staff, expertise, time, and equipment. It is basically used as a measure of how practical or beneficial the development of a software system will be to you (or organization). This activity recurs throughout the life cycle.

There are many types of feasibilities:

* Technical
* Operational
* Economic
* Schedule
* Specification
* Information
* Motivational
* Legal and Ethical

### **1.1.1 Technical Feasibility**

Our team is well skilled and motivated to complete the project. We have a good grip on all concepts of Javascript, React JS, Express JS and MongoDB. React Native being the hybrid technology will allow us to develop the application for both android and iOS with minimum effort and without requiring technical knowledge of both android and iOS.

### **1.1.2 Operational Feasibility**

The application is easy, users can use it, In order they will find it comfortable especially for those who are finding difficulties to come to the authority to solve their problems as payment also will be through the application.

The new system will not decrease the workforce as those employers will be responsible to take care of the users online and solve their problems rather than coming to the authority, legal and ethical issues must be considered in the new system in case something occurs.

**1.1.3 Economic Feasibility**

* **Cost Estimates:**

-  **Development Cost**

Organizations have sufficient numbers of systems which would be feasible for this application.

- **Maintenance Cost**

The cost of system maintenance represents a large proportion of the budget of most organizations that use software systems. More than 65% of software lifecycle cost is spent in the maintenance activities.

* **Benefit Estimates**

**-Tangible**

* Developing a new product to save time and money.

**- Intangible**

* Increase the number of clients that deal with Application.
* Customers will no longer spend hours waiting for availability of staff.
* Increase client’s happiness.
* Increase user satisfaction

### **1.1.4 Schedule Feasibility**

Project will be completed in the tenure of eight months and is divided into several deliverables which are to be submitted. Each deliverable will contain progress of the project. Development will be started in the last 3 months. For discussion sessions, several meetings are scheduled as well. This way will allow us to work in an efficient way. Individual work will be examined on a regular basis.

### **1.1.5 Specification Feasibility**

Our software will have the following specifications:

* Android App or iOS.
* Minimum 2 GB RAM or more.
* Processor: Pentium 4 or more
* Internet
* PC Hard disk: 120 GB
* Empty Storage 200 MB for mobile app.
* Any browser with JS support.

### **1.1.6 Information Feasibility**

We have enough information to complete this project. All the team members have done complete requirement gathering to get the required information to build this software. Moreover official documents of the university will help in developing a dynamic form portal i.e. one main objective of the application.

### **1.1.7 Motivational Feasibility**

* **Desirability:** The value of achieving the goal
* **Feasibility:** The likelihood that you can achieve the goal
* **Progress:** Your accomplishments as you work to reach your goal
* **Reward:** The payoff when you reach the goal

### **1.1.8 Legal & Ethical Feasibility**

Security measures will be taken to protect the information provided by students during Sign in the application.

Data processing systems must comply with the local data protection Acts. It must determine that any infringement and everything must comply with the legal requirements. Application will not reveal secrets of the university to prevent exercise of insider trading, etc. and to preserve the goodwill of the University.

## ***1.2 Project/Product Scope***

This is a data driven project i.e. it includes both mobile application and web app for the system that work independently. This application will facilitate the users especially students to submit applications to the university management online.The user will also be notified about the response from the management.This application will provide a chat bot that will assist the user with FAQs.

Moreover admin will be able to create dynamic forms as per user requirements. The whole project will grow in an incremental manner.

## ***1.3 Project/Product Costing***

A metric is some measurement we can make of a product or process in the overall development process. Metrics are split into two broad categories:

* Knowledge oriented metrics: these are oriented to tracking the process to evaluate, predict or monitor some part of the process.
* Achievement oriented metrics: these are often oriented to measuring some product aspect, often related to some overall measure of quality of the product.

Most of the work in the cost estimation field has focused on algorithmic cost modeling. In this process costs are analyzed using mathematical formulas linking costs or inputs with metrics to produce an estimated output. The formulae used in a formal model arise from the analysis of historical data. The accuracy of the model can be improved by calibrating the model to your specific development environment, which basically involves adjusting the weightings of the metrics.

**Cost estimation can be done by just one methodology.**

### **1.3.1 Project Cost Estimation By Function Point Analysis**

Function-oriented software metrics use a measure of the functionality delivered by the application as a normalization value. Since ‘functionality’ cannot be measured directly, it must be derived indirectly using other direct measures. Function-oriented metrics were first proposed by Albrecht, who suggested a measure called the function point. Function points are derived using an empirical relationship based on countable (direct) measures of software’s information domain and assessments of software complexity.

Function Point Analysis can provide a mechanism to track and monitor scope creep. Function Point counts at the end of requirements; analysis, design, code, testing and implementation can be compared. The function point count at the end of requirements and/or designs can be compared to function points actually delivered. If the project has grown, there has been scope creep. The amount of growth is an indication of how well requirements were gathered by and/or communicated to the project team. If the amount of growth of projects declines over time it is a natural assumption that communication with the user has improved.

Function points are computed by completing the table shown in the figure below. Five information domain characteristics are determined and counts are provided in the appropriate table location.

Information domain values are defined in the following manner:

**Number of user inputs:** Each user input that provides distinct application-oriented data to the software is counted. Inputs should be distinguished from inquiries, which are counted separately.

**Client:**

* Log in (Medium)
* Forgot Password (Low)
* Update Client (Low)
* Select Form (Medium)
* FAQ Chatbot (Low)
* Customise Form (High)
* Form filling (High)
* Request Approval (Medium)
* Log out (High)

**Admin:**

* Create Forms (High)
* Create Accounts (High)
* Delete Accounts (Medium)
* Log in (High)
* Log out (High)
* Respond to requests (Medium)

**Number of user outputs:**

* Output screens and reports (High)
* Notifications (Low)
* Answers to FAQ ( Low)
* Forms (Medium)
* Account settings (Low)
* Searchbar Response (Medium)

**Number of user inquiries:**

* Prompt and interrupts (Medium)
* Search Bar (Low)
* Feedback (High)

**Number of files:**

* Databases and directories (High)
* System data file (Medium)

**Number of external interfaces:**

* Printers Generate fee challan (Low)
* Databases (High)

Once these data have been collected, a complexity value is associated with each count. Organizations that use function point methods develop criteria for determining whether a particular entry is simple, average, or complex. Nonetheless, the determination of complexity is somewhat subjective.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Measurement**  **Parameter** | **Count** | **Weighting Factor**   |  |  |  | | --- | --- | --- | | **Low** | **Average** | **High** | | | | **Total** |
| **Number of user inputs** | 15 | 3 \* 3 = 9 | 5 \* 4 = 20 | 7 \* 6 = 42 | 71 |
| **Number of user outputs** | 6 | 3 \* 4 = 12 | 2 \* 5 = 10 | 1 \* 7 = 7 | 29 |
| **Number of**  **user inquires** | 3 | 1 \* 3 = 3 | 1 \* 4 = 4 | 1 \* 6 = 6 | 13 |
| **Number of files** | 2 | 0 \* 7 = 0 | 1 \* 10 = 10 | 1 \* 15 = 15 | 25 |
| **Number of external interfaces** | 3 | 1\* 5 = 5 | 0 \* 7 = 0 | 1 \* 10 = 10 | 15 |
|  | | | | **Count Total** | **153** |

**Value adjustment factor (Fi) Rating**

1. Data communications **4**

2. Distributed data processing **2**

3. Performance **4**

4. Heavily used configuration **3**

5. Transaction rate **1**

6. On-Line data entry **4**

7. End-user efficiency **4**

8. On-Line update **3**

9. Complex processing **3**

10. Reusability **2**

11. Installation ease **4**

12. Operational ease **4**

13. Multiple sites **0**

14. Facilitate change **2**

Sum of Value adjustment factor (Fi) = **40**

**FP est.=CountTotal\*[0.65 + 0.01 \*(Fi)]**

**FP est. = 153\* [0.65 + 0.01 \* (40)]**

**FP est. = 160.65 = 161 appr**

Where count total is the sum of all FP entries (153) obtained from above figure and (Fi) 40 is value adjustment factor (VAF) is based on 14 general system characteristics (GSC's) that rate the general functionality of the application being counted. Each characteristic has associated descriptions that help determine the degrees of influence of the characteristics. The degrees of influence range on a scale of zero to five, from no influence to strong influence.

Finally, Total Project Cost and Total Project Effort are calculated given the average productivity parameter for the system.

**\*Assuming: 1 month = 22 working days , per person**

**Average productivity** = 5 FP/month (per day: 0.23 FP)

**Cost/FP = labour rate / productivity parameter**

Cost/FP= 20000/23

**Cost/FP= 870 RS/FP**

**Total Project Cost = FP estimation \* (Cost/FP)**

Total Project Cost = 161\* 870

Total Project Cost = **140, 070** **RS**

**Total estimation effort = FP estimation / productivity parameter**

Total estimation effort = 161/23

Total estimation effort = 7pm. (7 \* 22 = **154 days**)

## ***1.4 CPM - Critical Path Method***

**Sequence Activities and Estimate Activity Co**

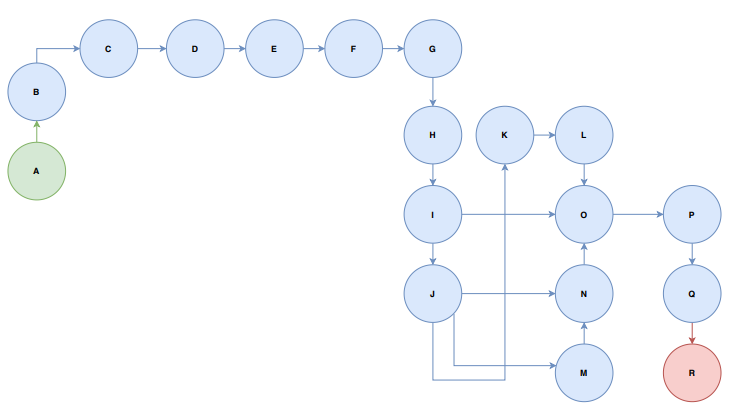
|  |  |
| --- | --- |
| **Activity Name** | **Symbol** |
| **Requirement Gathering** | A |
| **Proposal** | B |
| **Cost Estimation** | C |
| **Cost Analysis** | D |
| **Critical Path** | E |
| **Gantt Chart** | F |
| **DFD all Levels** | G |
| **Use case Diagram** | H |
| **Sequence Diagram** | I |
| **ER Diagram** | J |
| **Database Model** | K |
| **Creating Database** | L |
| **Development** | M |
| **Testing** | N |
| **Deployment** | O |
| **Maintenance** | P |

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity Name** | **Symbol** | **Duration(days)** | **Predecessor** |
| **Requirement Gathering** | **A** | **3** | **-** |
| **Proposal** | **B** | **2** | **A** |
| **Cost Estimation** | **C** | **2** | **B** |
| **Cost Analysis** | **D** | **2** | **C** |
| **Critical Path** | **E** | **1** | **D** |
| **Gantt Chart** | **F** | **2** | **E** |
| **Use Case Diagram** | **G** | **2** | **A, F** |
| **DFD all Levels** | **H** | **2** | **A,G** |
| **System Specification** | **I** | **3** | **H** |
| **ER Diagram** | **J** | **7** | **I** |
| **Sequence Diagram** | **K** | **2** | **J** |
| **Prototyping** | **L** | **10** | **K, H** |
| **Database Model** | **M** | **4** | **J** |
| **Creating Database** | **N** | **3** | **J, M** |
| **Development** | **O** | **60** | **N, L, I** |
| **Testing** | **P** | **8** | **O** |
| **Deployment** | **Q** | **3** | **P** |
| **Maintenance** | **R** | **15** | **Q** |

**6. Update CPM Diagram**

As the project progresses, the actual task completion times will be known and the network diagram can be updated to include this information. A new critical path may emerge, and structural changes may be made in the network if project requirements change.

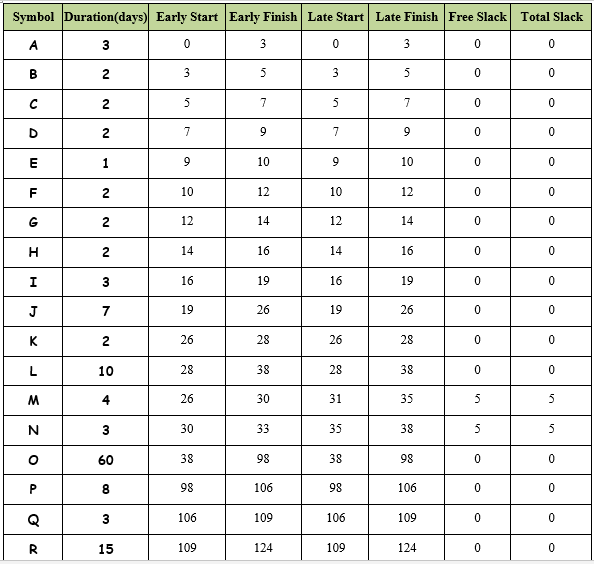
**CPM DIAGRAM**



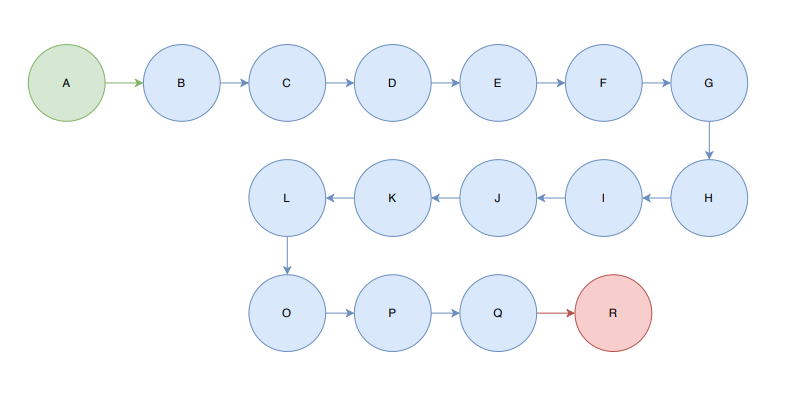
Network Diagram for the above-mentioned activities

***Critical Path Evaluation***

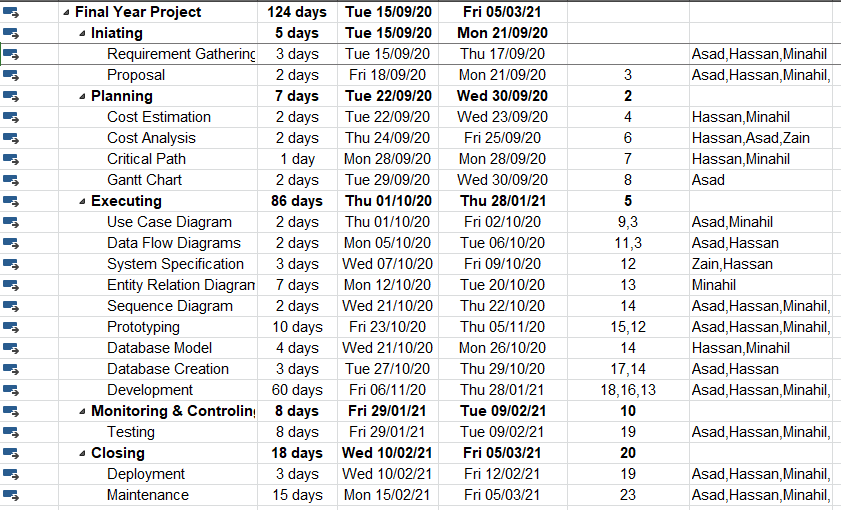
The parameters and slacks are calculated as follows:

****

**The critical path is:**

****

## ***1.5 Gantt chart***



## ***1.6 Introduction to Team member and their skill set***

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Member Information** | | **Skills Set** |
| 1 | **Asad Shabir** | **BSEF17M520** | **Specialization:** Full Stack Dev.  **Project relevant skills**: He is good in hybrid mobile application development as he worked on real time projects as part time in the market. Especially in JS frameworks like React.js and React Native. He has good team building and management skills. |
| 2 | **Minahil**  **Mustafa** | **BSEF17M518** | **Specialization:** Web Development and Management.  **Project relevant skills:** She has developed good skills in web development especially in JS frameworks like NodeJS, ReactJS, Express. Besides programming, she is also good at managing things and analytical skills. |
| 3 | **Muhammad**  **Hassan** | **BSEF17M541** | **Specialization:**  Front End and Information Technology  **Project relevant skills:** He is developing front end designs in HTML, CSS and JAVASCRIPTS effectively. He has a good hand in MERN Stack. Especially in JS Frameworks like React.js and React Native. |
| 4 | **Zain Ul Abideen** | **BSEF17M517** | **Specialization:** Back End and Database  **Project relevant skills:** Besides web development, he is good at designing databases e.g MongoDB. He has worked on extensible database structures. He has worked with NodeJS, Express framework of JS. |

## ***1.7 Tools and Technology with reasoning***

This project will use HTML, CSS, Bootstrap, Javascript, NodeJS, Express, MongoDB, React Js and React Native.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tools and**  **Techniques** |  | **Reasons** |
| 1 | **HTML** | It will be used to create web page structure. | |
| 2 | **CSS** | It is used to style the views of the site. | |
| 3 | Node.js | Javascript is the programming language used to code backend. As for backend server side of our website, we use Node.js. | |
| 4 | Microsoft Office | For documentation purposes. | |
| 5 | **Draw.io** | Draw.io is used for UML Diagrams | |
| 6 | MongoDB | Database used for retrieving and storing data for both mobile Application and website is MongoDB. | |
| 7 | **React Js** | React is a User Interface (UI) library. React is a tool for building UI components. We use it as the front-end language of our website | |
| 8 | Visual Studio Code | Visual studio code is used for website customization and event handling. | |
| 9 | Android Studio | Android studio is used for android applications | |
| 10 | Expo Client | Expo client for live testing android app | |

## ***1.8 Vision Document***

## Problem Statement:

## Large number of document forms are collected in the office on a daily basis but clients are not getting proper vision to track the process of their submitted form.

## Solution:

## This is an application that will allow the user of an application to view the track of form process digitally. He can then view the current situation of his process and can avoid going to the office again and again to ask about process completion.

## Actors:

## There are three main actors in this project.

## ● Client/User

## ● Head of Institute

## ● Administrator

## ***1.9 Risk List***

Following are some risks that have been identified in our project. The risks are given in decreasing order according to the severity. And this list will be updated later in the project as we face any other possible risks. The list in decreasing order of importance along with mitigation or contingency actions is as follows:

**1.** **Time Management:**

Scope and functionality of the project must comply with the schedules otherwise project completion may be delayed.

**Mitigation or contingency actions:** All the team members of the project must strictly follow the schedules.

**2.** **Technology Advancement:**

Technology is changing day by day. Advancement in technology may not accept the terms used in our project.

**Mitigation or contingency actions:** Develop the project considering maintainability aspects.

**3.** **Business Aspects:**

If the target audience is not aware of such a website, the website will be of no use.

**Mitigation or contingency actions:** Marketing of the project done properly building interest of restaurants.

**4.** **Team Members:**

Issues Whole work is divided into team members according to their skills. In case of any member’s absence in the meeting, there may be a difficulty in completing the project in time.

**Mitigation or contingency actions:** Team members are very strictly made to attend every meeting.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*REQUIREMENTS ENGINEERING\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## ***1 Introduction***

Requirements engineering process provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are transformed into an operational system. The task of capturing, structuring, and accurately representing the user's requirements so that they can be correctly embodied in systems which meet those requirements (i.e. are of good quality).

* Requirements elicitation
* Requirements analysis and negotiation
* Requirements specification
* System modeling
* Requirements validation
* Requirements management

Here, requirements specification is to be discussed. Requirements specification would lead to the following four steps:

* Identify external interfaces
* Development of context diagram
* Capture “shall statements
* Allocate requirements
* Prioritize requirements
* Development of requirements traceability matrix

### **1.1 Systems Specifications**

The following are the clauses that must be included while describing the system specifications.

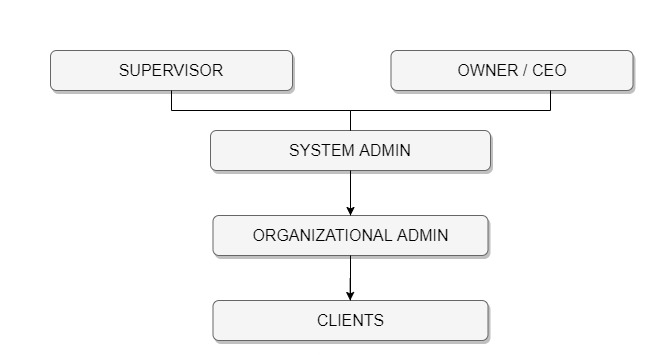
**Introduction**

A module that can manage all the activities related to different forms of any organization/Institution for instance in university’s forms like add/drop subject form, complaint form, feedback form, sick leave form, scholarship form, free concession form, Trip consent form, Society registration form, transcript generation form. Clients can login through credentials. After submitting their desired form, the form will be forwarded to the relevant administrator's inbox.

**Existing System**

There is no other system. This will be a standalone application that will manage forms for any organizations.

**Organizational Chart**



**Scope of the System**

This application will be used by clients and administration of any institution. Both will have a login id.

Client can:

* Submit their applications
* Customize form
* Send request for approval
* Take response from administrator
* FAQ chatbot
* Request new form
* Logout

Administration can:

* Create accounts for clients
* Delete Existing Account
* Can check client requests
* Give response to requests
* Create new forms
* Delete Existing forms
* Logout

**Summary of Requirements (Initial Requirements)**

The system should:

* Have a login & logout system
* Create new forms
* Delete existing forms
* Submit application
* Send requests
* Respond to requests
* Save all records
* FAQ chatbot
* Create accounts
* Delete accounts

**1.2 Identifying External Entities**

The Identification of External Entities is done in two phases.

**a. Over Specify Entities from Abstract**

● Client

● Administrator

**b. Perform Refinement**

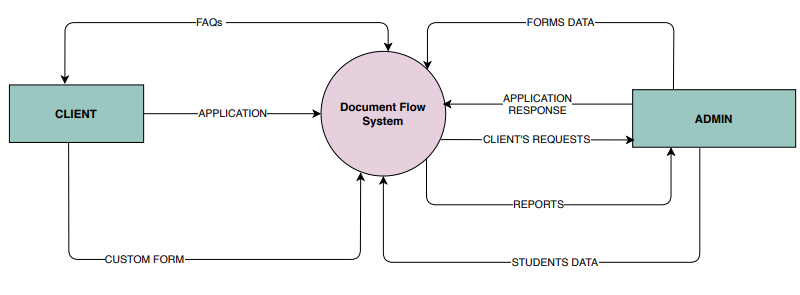
**●**  Internet

● Database

● Reviews

● Reports of Reviews

### **1.3 Context Level Data Flow Diagram**



### **1.4 Capture "shall" Statements**

|  |  |
| --- | --- |
| **External Entities** | **Initial Requirements** |
| Admin | Shall login on System |
| Admin | Shall create accounts for clients |
| Admin | Shall delete existing accounts |
| Admin | Shall check client requests |
| Admin | Shall give response to client’s requests |
| Admin | Shall create new form |
| Admin | Shall delete existing forms |
| Admin | Shall logout from the system |
| Client | Shall login to the system |
| Client | Shall submit their applications |
| Client | Shall customize forms |
| Client | Shall send requests to admin for approval |
| Client | Shall get response from management |
| Client | Shall apply for new forms |
| Client | Shall take help from FAQ chatbot |
| Client | Shall logout from the system |

### **1.5 Allocate Requirements**

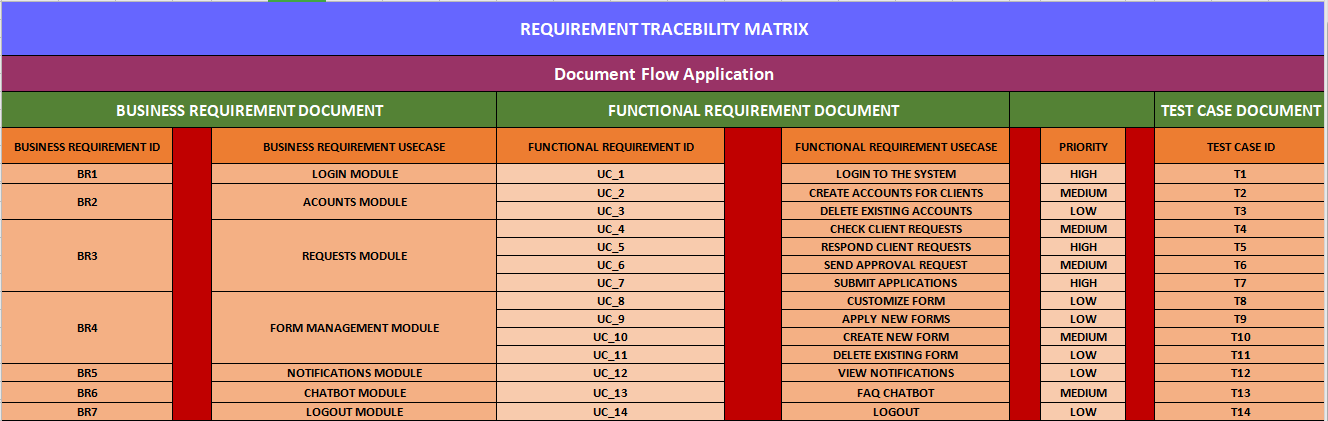
|  |  |  |
| --- | --- | --- |
| **External Entities** | **Initial Requirements** | **Use Case ID** |
| Admin | Shall login on System | UC\_1 |
| Admin | Shall create accounts for clients | UC\_2 |
| Admin | Shall delete existing accounts | UC\_3 |
| Admin | Shall check client requests | UC\_4 |
| Admin | Shall give response to client’s requests | UC\_5 |
| Admin | Shall create new form | UC\_6 |
| Admin | Shall delete existing forms | UC\_7 |
| Admin | Shall logout from the system | UC\_8 |
| Client | Shall login to the system | UC\_9 |
| Client | Shall submit their applications | UC\_10 |
| Client | Shall customize forms | UC\_11 |
| Client | Shall send requests to admin for approval | UC\_12 |
| Client | Shall view notification | UC\_13 |
| Client | Shall apply for new forms | UC\_14 |
| Client | Shall take help from FAQ chatbot | UC\_15 |
| Client | Shall logout from the system | UC\_16 |

### **1.6 Prioritize Requirements**

Requirements must be prioritized as this will help achieve tasks easily. Rank them as “highest, medium, and lowest”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Priority** | **External Entities** | **Initial Requirements** | **Use Case ID** |
| High | Admin | Shall log in the System | UC\_1 |
| Medium | Admin | Shall create accounts for clients | UC\_2 |
| Low | Admin | Shall delete existing accounts | UC\_3 |
| Medium | Admin | Shall check client requests | UC\_4 |
| High | Admin | Shall give response to client’s requests | UC\_5 |
| Medium | Admin | Shall create new form | UC\_6 |
| Low | Admin | Shall delete existing forms | UC\_7 |
| Low | Admin | Shall logout from the system | UC\_8 |
| High | Client | Shall login to the system | UC\_9 |
| High | Client | Shall submit their applications | UC\_10 |
| Low | Client | Shall customize forms | UC\_11 |
| Medium | Client | Shall send requests to admin for approval | UC\_12 |
| Low | Client | Shall View Notification | UC\_13 |
| Low | Client | Shall apply for new forms | UC\_14 |
| Medium | Client | Shall take help from FAQ chatbot | UC\_15 |
| Low | Client | Shall logout from the system | UC\_16 |

### **1.7 Requirements Trace-ability Matrix**



### **1.**8 H**igh Level Usecase Diagram**

