Final Report on

Paperless Catalog; A website hosted on AWS Platform

By

H.M Atahar Nur

Student ID: **1621192**

Spring, 2022

Supervisor:

Md Asif Bin Khaled

Lecturer

Department of Computer Science & Engineering

Independent University, Bangladesh

May 11, 2022

Dissertation submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

I attest that this report is my work, based on responsibilities that I have assigned to me during my internship period. In addition to that, I have acknowledged all material and sources used in this report.
I also certify that this report has not been submitted previously for any assessment in any other unit and that I have not forged the work of other students or persons.
However, following the internationally accepted academic guideline using other's written work is accurately cited if used in any part of this work.

Signature	Date
H.M Atahar Nur	
Name	

Acknowledgement

First and foremost, I desire to express my sincerest sense of gratitude to Almighty Al- lah because of His mercy and blessing that I have come so far. It has been a great privilege to work for Shadhin Lab LLC as an intern. I have received so much support and encouragement from the individuals of Shadhin Lab who have years of experience in Software Development. I also would like to thank the members of Shadhin Lab for spending their valuable time and knowledge and was essential for the completion of this report.

I express my gratefulness to my internal supervisor, Md Asif Bin Khaled, Lecturer, Department of Computer Science and Engineering, Independent University, Bangladesh (IUB), for his invaluable instructions, continuous guidance, support, and motivation during my internship period and preparation of this report.

I besides want to express my deepest gratitude to my external supervisor and my mentor Md. Masudur Rahman for his continuous support and guidance throughout the completion period of the project. His support and leading ability were the driving force of this project. I also want to thank my other project member Maruf Rahman Bhuiyan for his work on the frontend part of the project.

Finally, I proudly acknowledge the great sacrifices, good wishes, moral support, fruitful advice, inspirations, and encouragement from my family members, relatives, and friends.

H.M Atahar Nur
May 11, 2022

Letter of Transmittal

May 11, 2022
Md Asif Bin Khaled
Lecturer,
Department of Computer Science and Engineering
Independent University, Bangladesh
Subject: Letter of Transmittal for Internship Report, Spring 2022

Dear Sir,
This is to inform you that with due honor and respect, I, H.M Atahar Nur, ID - 1621192 from the internship course of Spring 2022 Semester, Section 11, would like to submit my Internship report. This report is based on my internship program and the project I have worked on. I have completed my internship at Shadhin Lab LLC which spanned from 11th January to 11th April

This report is based on my experience and the work I did at Shadhin Lab during my internship program. The primary goal for my internship was to gain knowledge by working in the software engineering industry and familiarize myself with all the different technology-related fields of the company, including research and development, documentation, software development, and to get acquainted with software development processes and practices. Throughout my internship at Shadhin Lab, I had to learn and adapt to the evolving technologies being used in different situations and requirements and to be able to apply them in real-life projects.

I shall be grateful if you are to receive this report and deliver your valuable judgment on this. I will be honoured if you find this report useful and informative to have an idea on the regarded issue.

Sincerely,
H.M Atahar Nur
ID - 1621192
Email: 1621192@iub.edu.bd

Evaluation Committee

.....

Signature

.....

Name

.....

Supervisor

.....

Signature

.....

Name

.....

Internal Examiner

.....

Signature

.....

Name

.....

External Examiner

.....

Signature

.....

Name

.....

Convener

Abstract

I carried out my internship at Shadhin Lab LLC. This internship opportunity gave me a chance to work with a real-life software application. This report is about the Paperless Catalog website. A website built using AWS Services like AWS Lambda, API Gateway, Cognito, S3, DynamoDB etc. The website is for restaurant owners and customer where the restaurant owner/admins can add, update and delete an menu item and publish the menu. On the other hand customers can view the menu of different restaurant simply visiting the website.I have joined Shadhin Lab as a backend web developer intern. It allowed me to gain raw firsthand experience of working in a corporate company, attend online scrum meetings, and develop projects for the company. It has allowed me to gain theoretical as wee as practical knowledge about AWS Services which ultimately resulted in this "Paperless Catalog" project. Therefore, this experience was challenging yet crucial as well, to help me elevate my skills and confidence for me to be ready to pursue my career in this line of work. The internship was of 3 months and the report mainly has been documented solely based on my overall learning and experiences from this course of time.

Keywords— AWS, paperless catalog, online food service

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview/Background of the Work

"Observability" describes how well we can understand what is happening in a system. This can be often done by collecting metrics, logs, or traces [1]. 'Paperless Catalog" is a website that is built using AWS Services that enables its user to interact with flawlessly. Its also uses AWS Cloudwatch to collect metrics (Data points) related to the app which help its admin user to pinpoint any error and debug the issues systematically. When an app is deployed for the users to use, there are some questions arise, like "Is the system up or down?" "Is it fast or slow as experienced by the end-users?". When a system is operating at a large speed and scale, we can't afford to be blind about the operations. We need to be able to answer a wide range of operational and business questions like these. We need to be able to spot problems as they arise even before they disrupt the customer experience, respond quickly, and resolve them as quickly as possible. "Paperless Catalog" is built using an array of interconnected services that help each other to achieve this goal. To achieve this insight, we need an observable system where all the metrics can be viewed and developers can see the strength of the system. This will help them to monitor, debug and maintain the system very easily.

1.2 Objectives

The main idea of the "Paperless Catalog" is to help restaurants owners to share their menu with customers online. Restaurant admins can add,delete and modify any item from the menu they have published once they are a valid user of the application. The customers then can view any restaurant menu without even have to sign in and can chat directly to a specific admin about their queries. To achieve this idea we had to adapt such a technology that is secured, robust and can scale at any time at any given situation. For this reason we have chosen to go with AWS Platform and its services. Using AWS services also would give us many other advantages. For example:

- **Detection**: Timely detection of a problem is the first step in observability. Ideally, this should be done before the problems affect the end-user. Detection should be proactive

and multi-featured, including alarms when performance gateways are breached, synthetic testing, and anomaly detection. A common performance metric is the mean time to detect (MTTD). [1]

- **Monitoring**: Monitoring tools keep the record of performance statistics over time so that usage patterns can be identified. Monitoring agents record selected metrics at set intervals and store the resulting data in a time-series format. Though the term "monitoring" is sometimes defined as different from observability, monitoring is an activity that makes a system observable, alongside other activities like tracing and logging. Monitoring along with tracing and logging are described as "three pillars of observability". [1]
- **Alerts**: When something goes wrong, we can get timely alerts through the observability tool. However, too-sensitive detection can lead to alarm fatigue, so alert management is also key to fine observability. [1]
- **Infrastructure monitoring**: Infrastructure monitoring lets us correlate metrics and logs from an infrastructure stack to understand and resolve the root causes of performance issues. [1]
- **Investigation**: Investigation is the most time-expensive phase of an operational event. When things are going wrong, it can be difficult to understand what is most important to fix. Multiple observability sources together can help us investigate quickly to understand the root cause, but to do this effectively you need to correlate data across metrics, logs, and traces. [1]
- **Cost Management**: All the AWS services charges in pay-per-use scheme. That means we would only have to pay storage and server charges the amount of services we are using.

1.3 Scopes

After the development of "Paperless Catalog", the scopes for its users would be:

- Fully functional, scalable, dynamic, and responsive website to interact with
- Real-time chat operation
- Error type detection whether it is a server-side or client-side error for admins
- Anomaly frequency observability
- AWS resource management for any AWS services (lambda, SQS, DynamoDB, etc)
- Graphical and visual overview of system statistics.

Chapter 2

Literature Review

2.1 Relationship with Undergraduate Studies

Special Topics in Computer Science (CSE490) is the course where I was introduced to the AWS platform. From this course, I have learned about AWS as a Platform as a Service (PaaS) and its many services like AWS Lambda function, API Gateways, DynamoDB, SQS, SNS, Cloudwatch, Serverless application model (SAM), and many more. The knowledge gathered from this course is directly connected to my whole internship journey and also my final project. From **Cloud Computing (CSE 472)**, I've learned about cloud as a service in general, and many cloud service providers like Amazon, Google, Microsoft etc. From this course I've leaned how the cloud works as a service and how we can use cloud as a platform, infrastructure or even a software to run our businesses. **Object-Oriented Programming (CSE 213)** course has taught me about classes and their objects of in programming. It also taught me how to write modular programs to make codes less redundant and more reusable. As the framework works the modular way it helped to code faster. In **System Analysis and Design (CSE 307)** I have learned how to manage the software development process from this course. This course also taught me about many aspects of software development like Software Development Life Cycle (SDLC), flow diagrams, activity diagrams, UMLs, Functional and Nonfunctional requirements, and various methodology which helped me to design projects. **Web Application and Internet (CSE 309)** is the course where the development of web applications was taught. It covered very important technologies that are highly demanding in the industry, such as HTML, CSS, JavaScript, PHP, MySQL, API. The tools and technologies learned from this course immensely contributed to understanding the front-end of the websites.

2.2 Related Services

There are many services that currently are operational related to my project work. AWS has many of its own native observability solutions that have been developed from the ground up to observe other AWS services, to operate at a cloud scale, and to provide enterprise-level security as well as many other cloud service providers.

- **Amazon CloudWatch**: Amazon CloudWatch automatically ingests operational data from AWS applications and applies machine learning models informed by years of Amazon.com and AWS operational excellence to identify anomalous application behavior and surface critical issues before they cause any disruptions.
- **AWS X-Ray**: AWS X-Ray perform distributed tracing across multiple applications and systems to help find latency in a system and target it for improvement.
- **Amazon CodeGuru**: Amazon CodeGuru Automatically ingests operational data from AWS applications and applies machine learning models informed by years of Amazon.com and AWS operational excellence to identify anomalous application behavior and surface critical issues before they cause outages or service disruptions.

2.3 Related Service Users

There are many companies who are operating their business through the AWS platform. They are also using AWS Services to monitor and maintain their systems. Some companies are listed below:

- **Mapbox**: Mapbox is an open-source mapping platform for custom-designed maps that reaches more than 300 million people each month. Mapbox uses Amazon CloudWatch for ingestion of multiple data sources including native AWS metrics, custom metrics, and logs as well as monitoring and visualization of key workloads and resource optimization.
- **PushPay**: Pushpay's purpose is to bring people together by strengthening community, connection, and belonging. Pushpay build world-class giving and mobile app publishing solutions to help organizations grow their communities. Pushpay is using CloudWatch Logs Insights to query logs within CloudWatch Logs reducing operational complexity.
- **SendGrid**: SendGrid is a provider of cloud email and sends more than 40 billion emails each month for more than 69,000 paying customers. SendGrid adopted Amazon CloudWatch early in its migration to AWS in order to gain system visibility, operational insights, and resource optimization.

Chapter 3

Project Management & Financing

3.1 Work Breakdown Structure

The Work Breakdown Structure (WBS) is a visual, hierarchical, and deliverable-oriented method that helps to complete complex, multi-step projects. It applies the divide-and-conquer paradigm to large projects, so tasks are completed faster and more efficiently. It is the hierarchical tree structure-property that outlines a project and breaks it down into more inadequate pieces. The goal of WBS is to make large projects more manageable by breaking down the complete project into smaller parts. Team members can work on different features at the same time, which ultimately leads to better productivity. We utilize the top-down approach to appear Work Breakdown Structure (WBS) to this extend. Using WBS will be effective for us to protect work quality and easy to synchronize the whole project. The WBS of "Paperless Catalog" project is given below:

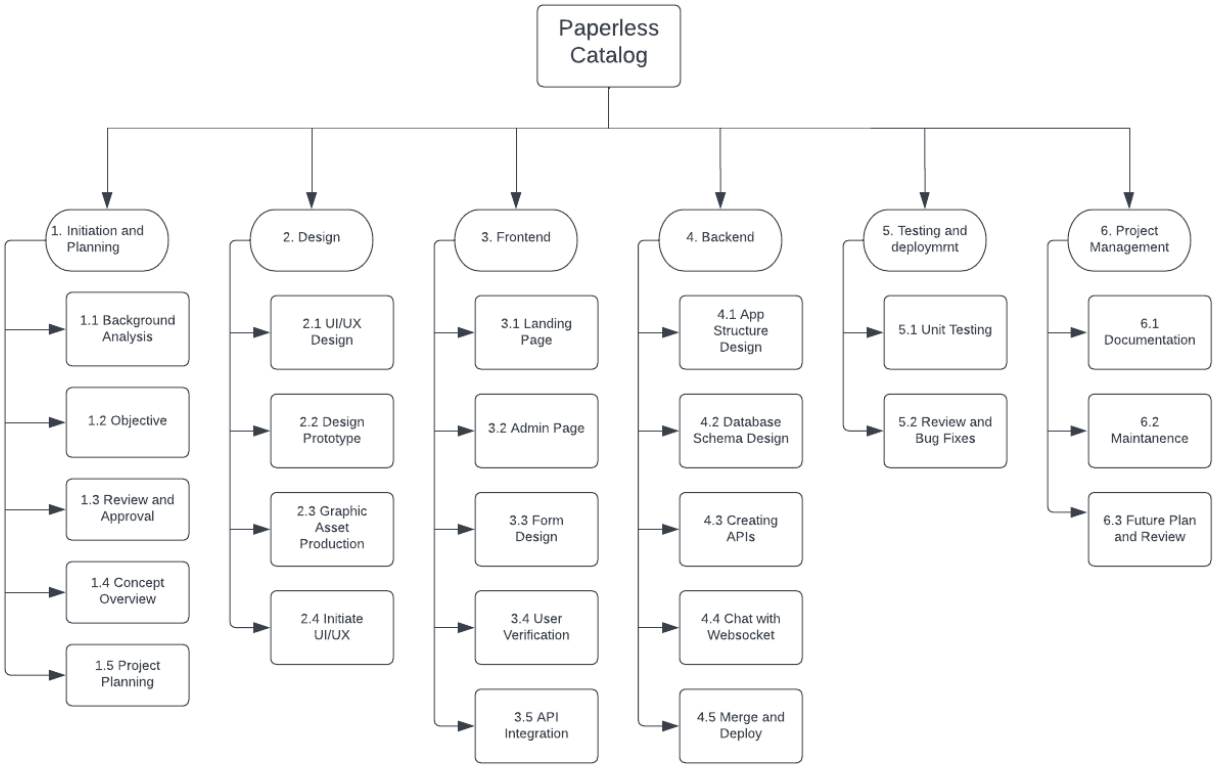

Figure 3.1: Work Breakdown Structure of Paperless Catalog

3.2 Process/Activity wise Time Distribution

All the activities related to this project are listed in the WBS. We as a team attempted to complete these tasks in a given time. The time for completing the project is estimated at 90 days.

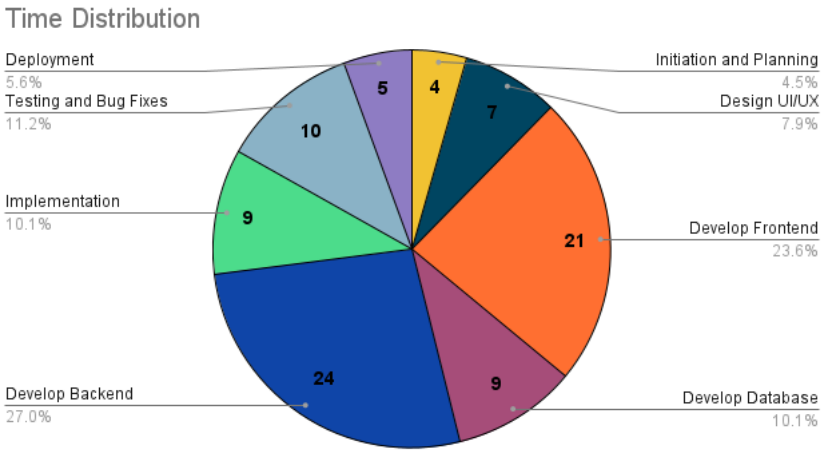

Figure 3.2: Time Distribution of the Project

3.3 Gantt Chart

A Gantt Chart is a graphical presentation of a project schedule, commonly used in project management and is one of the most popular and beneficial techniques of showing activities (tasks or events) displayed against time. It helps to estimate how long the project should take, determine the resources needed and plan in the order in which tasks will be completed. Work breakdown structure appears as the sum of activities and to complete these activities a certain time is required which is portrayed within the Gantt Chart. It is also useful to monitor the project's advancement once it has begun. It helps to have a more precise vision of what should have been delivered by a certain time frame and when the project falls behind schedule; proper actions are taken to bring it back to track.

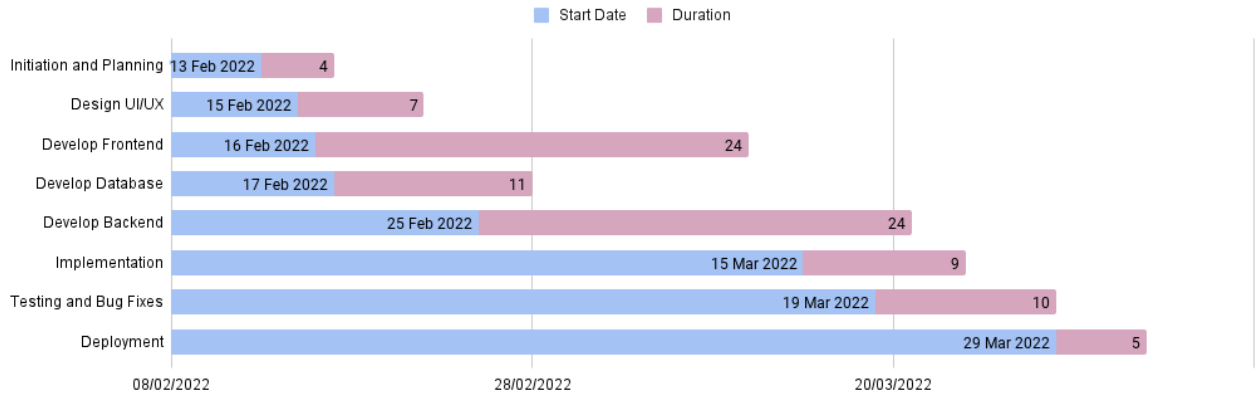

Figure 3.3: Gantt Chart of Paperless Catalog

3.4 Estimated Costing

As my application will be hosted on AWS Platform, some resource costing is inevitable. Luckily i was given permission to use the Shadhin Lab's cloud hosting platform. The AWS services are billed on pay-per-use schema so it is somewhat difficult to project the exact costing before we complete the whole project. But the approximate cost was evaluated around nearly 30,000 BDT. It can be expanded on the changes in the project.

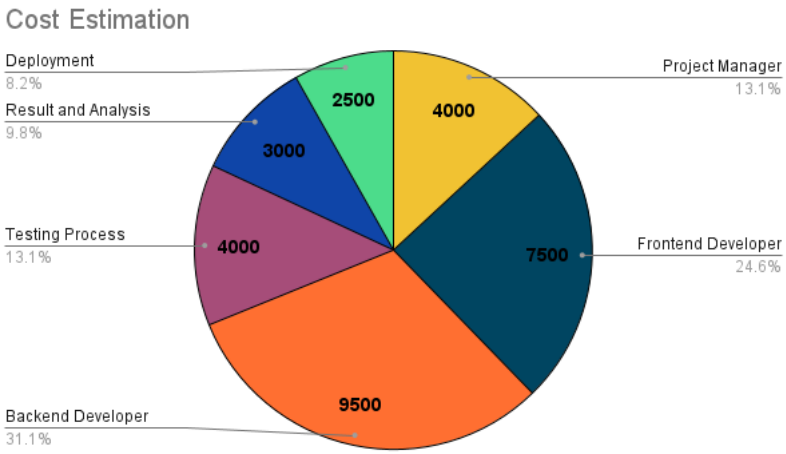

Figure 3.4: Cost Estimation of the Project

Sector	Cost
Project Manager	4000 Taka
Frontend Developer	7500 Taka
Backend Developer	9500 Taka
Testing Process	4000 Taka
Result and Analysis	3000 Taka
Deployment	2500 Taka

Table 3.1: Cost Estimation

Chapter 4

Methodology

4.1 Agile Methodology

Agile methodology is a practice that promotes continue interaction of development and testing during the SDLC process of any project. Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment. Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly. All of these builds are provided in iterations, and each iteration lasts from one to three weeks. The Agile SDLC model separates the product into cycles and delivers a working product very quickly. This methodology produces a succession of releases. Teams use the agile development methodology to minimize risk (such as bugs,cost overruns, and changing requirements) when adding new functionality. In all agile methods, teams develop the software in iterations that contain mini-increments of the new functionality. Agile is an umbrella term for several methods and practices.

Chapter 5

Body of the Project

5.1 Work Description

During my internship, i worked on an application called "Paperless Catalog". Its basically a website for restaurants to share their menu with all the customer. The application is built using the AWS services and also hosted on AWS platform. The current version of the website has limited interactions with the users. Currently, the customers can register and log in to the website and view the list of menu shared by various restaurants, they can also view the details of a specific menu item. On the Admin side, there is also a limited number of features. For example, admins of the restaurants can register and log in to the website and they can monitor their menu in the website by adding, updating and removing the menu. They can also monitor user interactions with their registered restaurant. REST APIs are used for the dynamic feature of the website.

5.2 System Analysis

Systems development is a systematic process that includes phases such as planning, analysis, design, deployment, and maintenance. System analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted to study a system or its parts to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do. This chapter contains parts of System Analysis that will help understand the project better. System analysis is important because it provides an avenue for solutions in the system through various tasks involved in doing the analysis. [2]

5.2.1 Six Element Analysis

The Six Element Analysis of "Paperless Catalog" is given below: 5.1

Process	Human	Non Computing Hardware	Computing Hardware	Software	Database	Network
Sign up	Both	N/A	PC, Laptop, Mobile	Any Browser	DynamoDB	WAN
Login	Both	N/A	PC, Laptop, Mobile	Any Browser	DynamoDB	WAN
Add Menu	Admin	N/A	PC, Laptop, Mobile	Any Browser	DynamoDB	WAN
Update Menu	Admin	N/A	PC, Laptop, Mobile	Any Browser	DynamoDB	WAN
Delete Menu	Admin	N/A	PC, Laptop, Mobile	Any Browser	DynamoDB	WAN
View Menu	Both	N/A	PC, Laptop, Mobile	Any Browser	DynamoDB	WAN

Table 5.1: Six Element Analysis of "Paperless Catalog"

5.2.2 Feasibility Analysis

Feasibility analysis is a study to assess the feasibility of a proposed project or system. It is an introductory survey for the system's examination. It intends to provide information to help a later in-depth inquiry. It is a measurement process of the software product in terms of how useful the developed product will be for the business from a functional point of view. A feasibility study is conducted out based on many purposes to examine whether software features will be right in terms of development, implantation, the contribution of the project to the organization, etc. The report produced at the end of the feasibility analysis includes suggestions and reasonable arguments. The report will be helpful for the management to decide whether they should commit further resources to the proposed project.
There are various measures of feasibility that helps to decide whether a particular project is feasible or not. Main parts of feasibility study:

- **Technical Feasibility:** In technical feasibility, present resources both hardware and software along with required technology are analyzed to develop the project. The technical feasibility analysis gives a report on whether there exist correct required resources and technologies which will be used for the project development. It also evaluates the technical skills and capabilities of the technical team and it also analyzes whether current technology can be used or not. "Paperless Catalog" website is built using AWS Services like AWS Lambda, Cognito, API Gateway, DynamoDB, AWS S3, Serverless Application

Model(SAM) etc. These are the technologies that are very popular in the modern software industry because of the nature of serverless platform is "highly scalable" and everyone involved in the making of this project had the skills to work with these technologies mentioned above. Shadhin Lab LLC has skilled technical experts to complete and maintain this project and also has the resource to deploy the application to the cloud. Shadhin Lab LLC also trained and guided me to complete this project with accuracy. Hence, it can be concluded that the project is technically feasible.

- **Operational Feasibility:** In operational feasibility degree of providing service to requirements is analyzed along with how easy the product will be to operate and maintain after development. An application must be user-friendly as well as it must work with no difficulties. Along with these, other operational scopes are determining the usability of the product and determining whether a suggested solution by the software development team is acceptable or not. "Paperless Catalog" is a simple website built with complex technology. Users can easily navigate through the whole website as it is self-explanatory. Our project is developed from scratch using modern web development technologies, frameworks and it is highly scalable no matter how the user load is. It will also allow us to customize our project to our perception. As a result, the project can be called operationally feasible.

- **Economic Feasibility:** In economic feasibility analysis, the cost and benefit of the project are analyzed. It is a detailed study where the cost of the project from all of the different spaces are combined. For example, the cost for required hardware and software resources, UI/UX design and development cost, operational cost, etc. And then it is calculated whether the project will be financially beneficial for the company or not. The only service that is needed for the development of "Paperless Catalog" is just a cloud server, and our application is hosted on AWS platform. The pricing scheme of which is "pay-per-use". That means the cost of our hosting is depends on the usage of our application. So the proposed system is justifiable in terms of cost and benefit and it ensures that the investment in this system provides a reasonable return. In conclusion, the project can be called economically feasible.

5.2.3 Problem Solution Analysis

The Problem Solution Analysis table is given below: 5.2

Problem	Analysis	Solution	Constraints
Data Loss	Any time data can be lost for various reasons (Virus attack, attack by hackers, unfortunate system crash)	Database system is more secure and deleted, lost data easily can be restored	Internet Connectivity
Time Complexity	accessing a specific information requires a lot of time to find and compute	Database system is easily accessible and need less time	Internet Connectivity
Human Error	Invalid user inputs	Form validation from both client side and server side	N/A
Analysis the data	Some problem occurs due to fetching manually different information	Database system can collect all the data and it is very easy to analyses the data	Internet Connectivity
Steal Data	Data can be stolen	Authentication	Internet Connectivity

Table 5.2: Problem Solution and Constraints Analysis Table

5.3 System Design

System design is the process of designing the elements of a system such as the architecture, modules, and components, the different interfaces of those components, and the data that goes through that system. System design is deriving a solution which satisfies software or system's requirement. We can define software design as translating requirements into software components and interactions among them. Any design may be modeled as a directed graph made up of entities with attributes that participate in relationships. A design represents the system, how it will work, and how it can be assessed for quality. Design is the way to translate a client's requirements into a system or software product accurately. Software architecture provides an abstract representation of the overall structure of software. This chapter contains numerous design-level diagrams to have a clearer understanding of the system and flow of data. [3]

5.3.1 Rich Picture

A rich picture is a way to explore, acknowledge and define a situation and express it through diagrams to create a preliminary mental model. A rich picture helps to open discussion and

come to a broad, shared understanding of a situation. It comprises pictures, content, images, and symbols, which are all utilized to demonstrate the graphically the circumstance. A rich picture illustrates how the application works. The process beings with the user turning on the application on their device using a browser while being connected to the internet. Then they can roam around the website to get their necessary information. They can choose to view the details of a particular menu item. They can also choose the send messages to the admin via the chat application. The admin can do their part after logging in to the application.

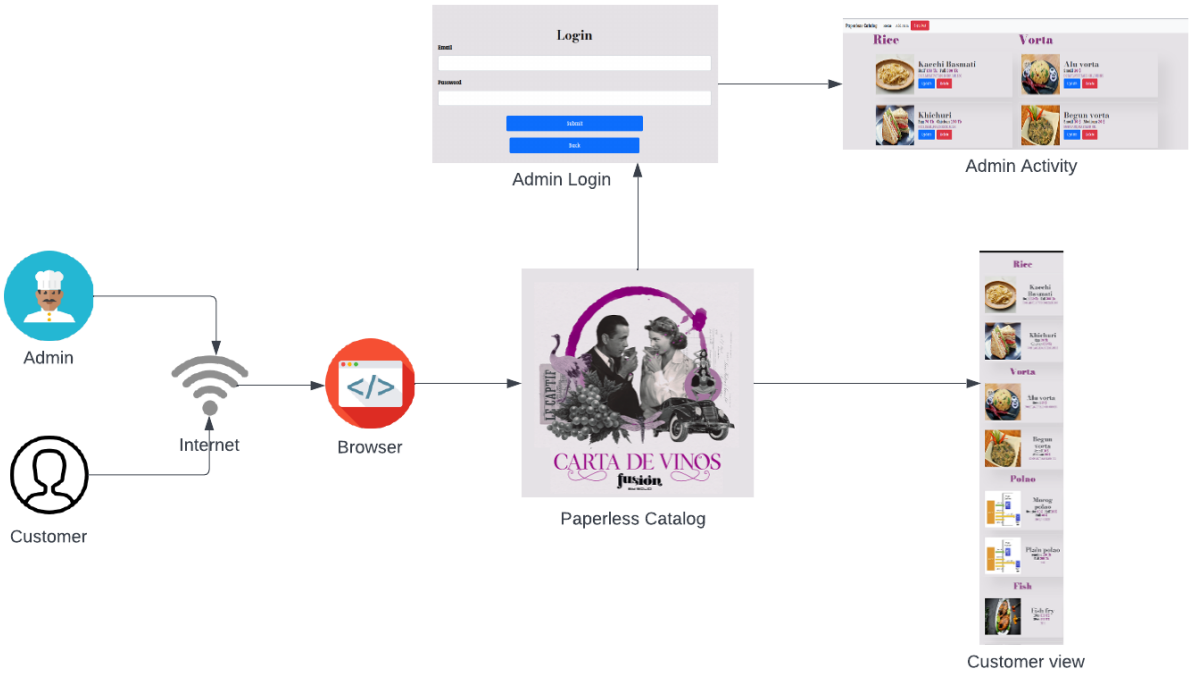

Figure 5.1: Rich Picture of Paperless Catalog

5.3.2 UML Diagrams

UML, which stands for Unified Modeling Language, is a way to visually represent the architecture, design, and implementation of complex software systems. When we are writing code, there are thousands of lines in an application, and it is difficult to keep track of the relationships and hierarchies within a software system. UML diagrams divide that software system into components and sub-components. Use case diagram model shows how users, displayed as stick figures called "actors," interact with the system. The use case diagram of Paperless Catalog is given below:

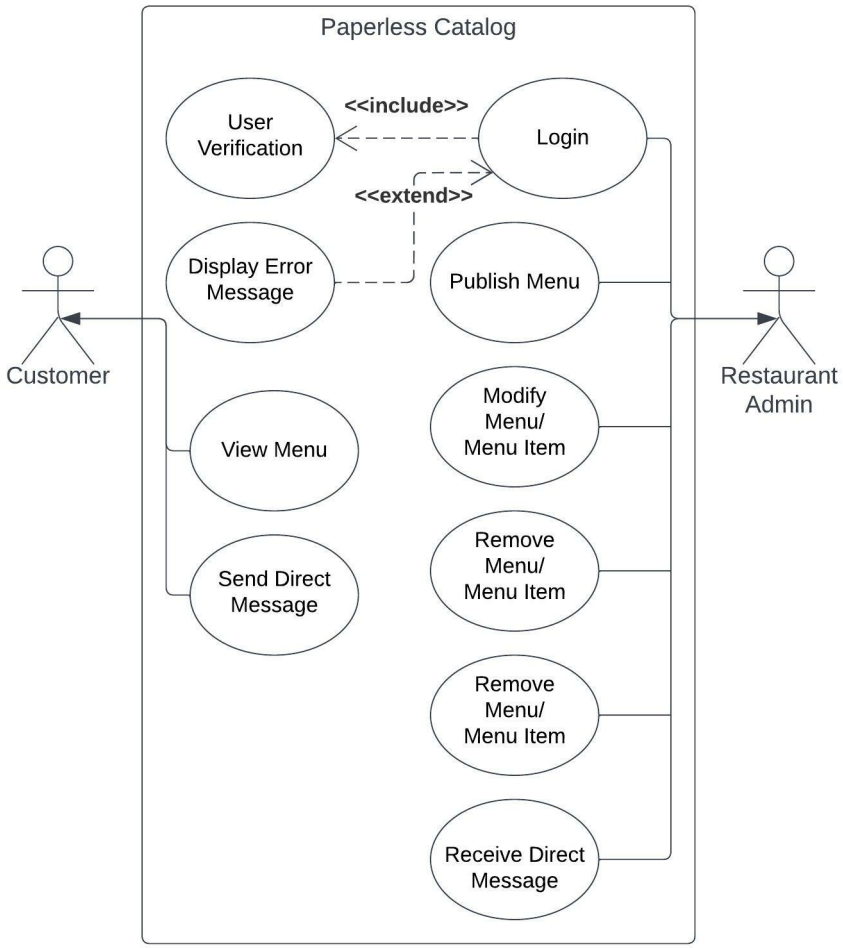

Figure 5.2: Use Case Diagram of Paperless Catalog

5.3.3 Process Flow Diagram

A process flow diagram shows us the overall overview and state of a process from its start to finish. It help us to visualize the different state and decision points of a process and how the underlying logic of our process will be based on. The process flow diagram for restaurant admin is given below:

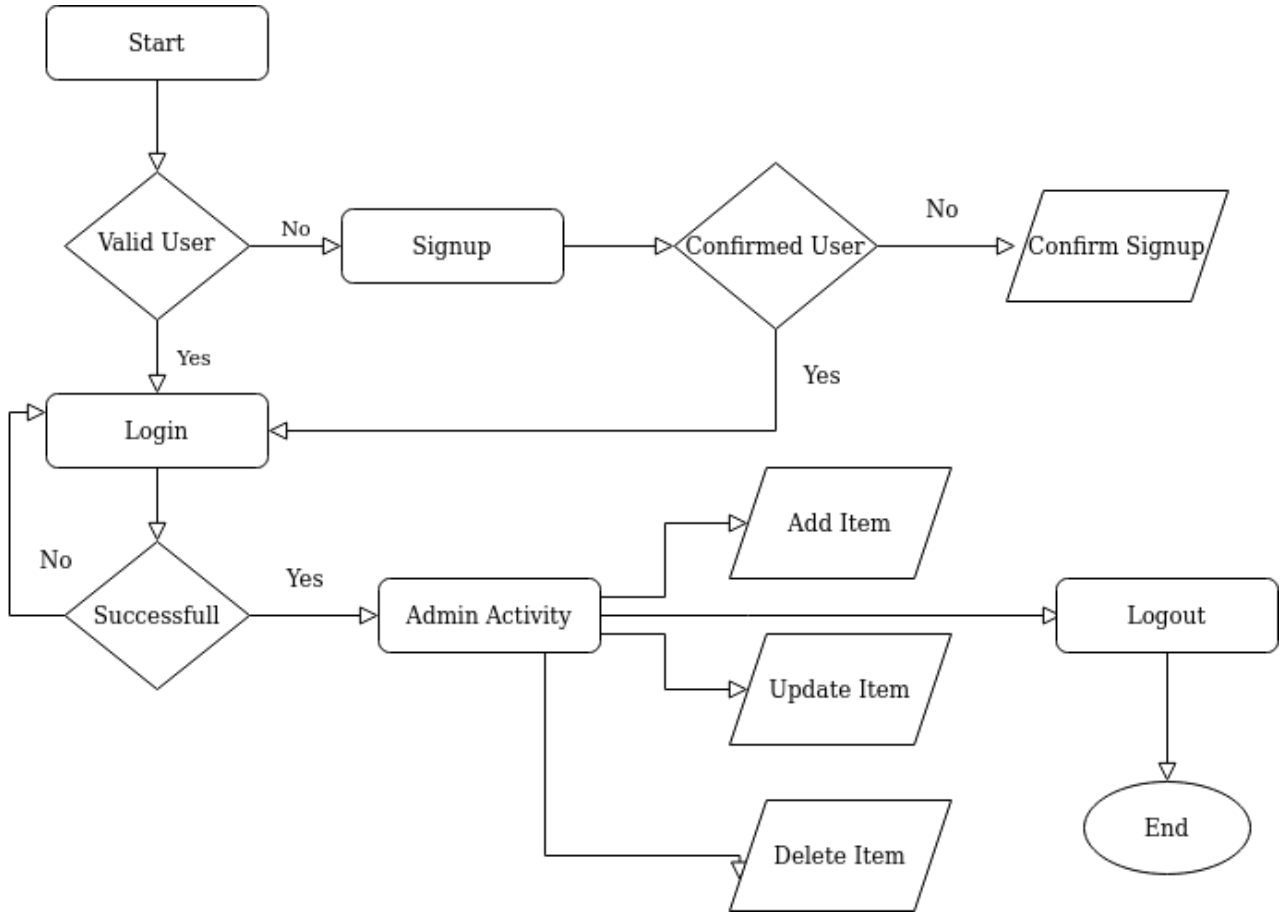

Figure 5.3: Process Flow Diagram of Admin User

5.3.4 Requirement Analysis

Requirement analysis involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore, requirement analysis means analyzing, document, validate and manage software or system requirements. Great quality requirements are documented, actionable, measurable, testable, traceable, helps to identify business opportunities, and are defined to facilitate system design as well.
The software requirements are descriptions of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from the client's point of view. Requirements can be divided into two types; functional and non-functional requirements.

5.3.5 Functional Requirements

The functional requirement typically species something a system should do. Functional requirements are features that must be included in an information system. Functional requirements help to satisfy the business requirement and it needs to be acceptable to the users. Functional requirements are the operations and activities that a system must be able to perform. Functional requirement defines applications nature and components as well as what these components are supposed to accomplish. The following functional requirements are assembled using requirements gathering methods. The inputs, process, output, pre-condition and post-condition for functional requirements are discussed below:

Function: Must be scalable at any moment		
Input: N/A	**Process**:Application must be developed in a scalable environment	**Output**:Application will scale at any moment at any given number of requests at a time
Pre-condition:User must have internet enabled computer or smart device.		
Post-condition: Application will process bulk requests		
Alternative Option:N/A		

Table 5.3: Functional Requirement 1 - Scalability

Function: Must be compatible with any kind of device		
Input: N/A	**Process**:Application must be developed in a common development environment	**Output**:Application can be accessed by any kind of device
Pre-condition:User must have internet enabled computer or smart device.		
Post-condition: Everyone can use the application		
Alternative Option:N/A		

Table 5.4: Functional Requirement 2 - Compatibality

Function: Sign Up		
Input: Valid Email and password	**Process**:Admin Sign up	**Output**:Confirm sign up interface
Pre-condition:User must have internet enabled computer or smart device.		
Post-condition: Admin can log in and use the application		
Alternative Option: If any error occurs showing relevant error message		

Table 5.5: Functional Requirement 3 - Sign up

Function: Login		
Input: Valid Email and password	**Process**:Admin Login	**Output**: Admin interface of the application
Pre-condition: User must confirm sign up		
Post-condition: Admin can log in and use the application		
Alternative Option: If any error occurs showing relevant error message		

Table 5.6: Functional Requirement 4 - Login

Function: Administrative activity		
Input: N/A	**Process**: Admin can publish, modify, delete menu	**Output**: Admin interface of the application
Pre-condition: Must be logged in and have internet enabled smart phone or computer		
Post-condition: Admin can maintain and monitor the application		
Alternative Option: N/A		

Table 5.7: Functional Requirement 5 - Admin activity

Function: Real time chat		
Input: Valid Email, password and username	**Process**: Chat with customers	**Output**: Information about the menu
Pre-condition: Need of information		
Post-condition: Customer must wait for some admin to respond		
Alternative Option: N/A		

Table 5.8: Functional Requirement 6 - Chat

5.3.6 Non-Functional Requirements

A non-functional requirement is a qualitative requirement for a project. It judges the software system based on responsiveness, usability, security, portability, etc. Non-functional requirements are described below:

- **Reliability:** The system is well trusted. Admin and user can use this software very easily. The software is made so simple that users can understand the interface and procedure very clearly. The system is built with the latest web technologies, well maintained and scalable, so the application is reliable.
- **Performance:** The main purpose of building this simple website with complex frameworks is to gain extra performance. It is required to exhibit and to meet the user's need. It describes the acceptable throughput rate and a satisfactory response time. Users tend to spend more time on the website when it loads faster, and this website should provide a smooth experience to all kinds of users.
- **Security:** Security requirements are another leading type of non-functional requirement. All the information from the client-side to the server-side is secured. The website deals with a limited number of data but the architecture follows all the latest security measurements. Only admins can perform administrative tasks and appointed developers have access to the core code. The back-end of this application is built with AWS Lambda, Api GateWay, Cognito, DynamoDB, S3 like AWS services they have several levels of security. So the system is well secured.

5.4 Architecture

The architecture of a system describes its major components, their relationships (structures), and how they interact with each other. Software architecture is what defines and structures a solution that meets technical and operational specifications. It serves as a blueprint for a system. It provides an abstraction to manage the system complexity and establish a communication and coordination mechanism among components. Software architecture optimizes attributes involving a series of decisions, such as security, performance, manageability. Many types of architecture are used in the software industry. We are using client-server-architecture for our system.

5.4.1 Client-Server-Architecture

The client-server architecture is a distributed system structure that partitions task or workload between the providers of a resource or service, called a server, and service requests called clients. Distributed systems are where the system or software runs on a loosely integrated group of cooperating processors linked by a network. It means a set of separate devices that are capable of autonomous operation, linked by a network. In client-server architecture server hosts, manages, and delivers most of the resources and most of the services are consumed by the client.

It is a type of architecture where one or more clients are connected to a central server over the internet connection. Clients do not share any of their resources. When the client device sends a request for data to the server, the server accepts the requested process and delivers the data packets requested back to the client.

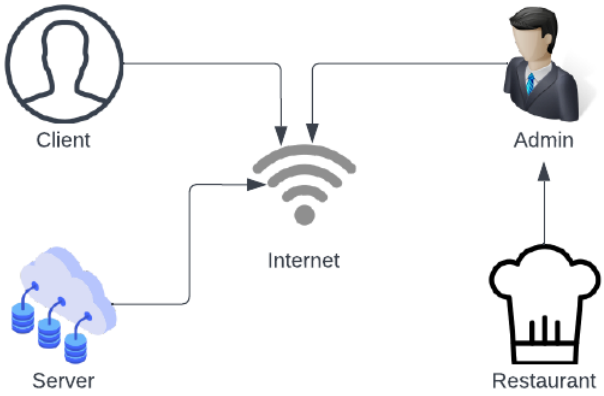

Figure 5.4: Client-Server Diagram of Paperless Catalog

Chapter 6

Result and Analysis

While working on a project, we always try to big our best to get the best possible outcome. If we don't get desired result, all the hard work will to in vain. To get the best result, at first, we figure out the bugs and fixed it. Because without fixing it, we can't move on. This web-based application was built to satisfy the users, to save their time as well as solve their problem. The front-end part was designed in a way so that people from all class can use this software easily. They don't need much technical knowledge to use it. Users can easily find the products they are looking for and they can get their order by staying at home. So, we tried to design the interfaces which will attract the customers. User can easily view products from various categories. Moreover, users can provide their view on products, delivery system etc. If they are having any issue, they can also inform the company.

Chapter 7

Project as Engineering Problem Analysis

7.1 Sustainability of the Project/Work

The sustainability of the product refers to its ability to be managed and refreshed. In this modern world, new applications are released every day, and these applications need to be maintained and continuously updated for their user base.

The use of smartphones and the internet have drastically increased among all classes of people in Bangladesh. The growth suggests that it would increase even further in the future. Community sustainability means how much and how actively the users will support the project. Support comes in many forms, such as visiting the website, giving feedback, refereeing to other people, etc. Paperless Catalog is a web app that currently only has a web version so the website is just a communication medium between the users and the service providers. Paperless Catalog is built using AWS services and a serverless application that is hosted in AWS platform. So no matter the user load is, the application is able to scale at any point. As the user of the app grows, all of them will be able to experience a smooth, effortless interaction with the app which will draw attraction. So, as the users will grow, it can be said that it is sustainable in terms of community.

A project is financially sustainable when the project's running costs are maintained by the revenue generated by the product that the project sells. An application's running cost includes - server cost, database storage cost, third-party API cost, etc. As Paperless Catalog will be used as an utility service of a company, adding this website only brings more users which will ultimately lead to more revenue generation. Thus, the project can be determined as financially sustainable.

Organizational sustainability is related to how the organization will continue to operate the release of the application. The Paperless Catalog is a web app that currently has only a web version. The main goal of this project is to make communication and information medium for the users. Since Paperless Catalog is serverless,it can be said that the project on the web will only grow as more and more organizations realize the advantages of serverless applications. In

conclusion, it can be said that the project is organizationally sustainable.

7.2 Social and Environmental Effects and Analysis

Websites are popular among people for the characteristics they offer to the users. Websites make interaction with people quite more straightforward. People experience a lot of advantages of using websites in different forms of their everyday work. Some benefits of having a website are - increased customer base, increased accessibility, easy access to information, continuous fresh content, gaining clients, securing brand online, and much more. The aim of the "Paperless Catalog" website is simple to reach more people, get them to know about the app, and also lower the boundary between the users and the app. As burgeoning industry, serverless applications are clearly on the rise. From start-up company to big companies, everyone is now moving their business to the cloud and welcoming the serverless framework for its many traits like scaling, sustainability, minimalist cost effectiveness etc.

7.3 Addressing Ethics and Ethical Issues

In today's world, there are an immense amount of data generated every day, which sometimes ends up with data collection, hacking, cyber crimes, etc. And some rules and ethics are needed to be followed when working on an application. Since the "Paperless Catalog" website is currently the only informative website, we believe that the application does not breach any code of conduct of application release and development since they all have been taken into a serious concern. At the "Paperless Catalog" website, there is only a collection of limited and relevant user data. The website collects the data strictly related to queries and requests for the menu item; other than those no other data is neither collected nor stored, even the chat data are not being stored. The website only collects limited amounts of data if the user interacts. The website does not let any service, any application, or any third party have access to the collected data. So there is no data sharing or selling.

Chapter 8

Lesson Learned

8.1 Problems Faced During this Period

As the AWS platform was totally new to me, I have faced many issues to learn and practice the tasks that I was assigned. Also, I had to learn many new libraries like pydantic, boto3 in order to complete the project. While working on my chat feature of the project, i faced some issue because i hadn't got any previous knowledge on websocket API's. During my internship period i have learned and worked with numerous AWS services like

- AWS Lambda
- AWS API Gateway
- AWS Cognito
- AWS SQS
- DynamoDB
- Cloudwatch

For most of the part i have used python as the programming language to implement the functionality of my project and as a medium to use all the services. For that to happen i also had to learn new libraries like

- pydantic
- boto3
- requests

8.2 Solution of those Problems

The Special topic (CSE490) course really helped me a lot to get a head-start on this project. Also, i had to invest a lot of time to learn and practice on these new technologies. To overcome

the difficulties i spent more and more time on research and fact finding. Also most of my time was spent on reading documentations of different libraries and blogposts published in the internet for different error i have encountered.

Chapter 9

Future Work & Conclusion

9.1 Future Work

This project "Paperless Catalog" website is just the beginning and there are numerous mapped features that are to be added in the imminent future. The whole idea of making this website with the latest front-end technologies is to implement highly scalable web app.

9.2 Conclusion

During my internship at Shadhin Lab LLC, I worked on various AWS services and implemented my knowledge to develop a web application called "Paperless Catalog". Currently, the website only deals with user queries and user requests for the menu item to view and realtime chat with the service providers. Working in Shadhin Lab as an intern has been a wonderful experience. I have discovered a lot about developing different kinds of applications and also about development methods. Working with cutting-edge technologies like python, AWS Lambda,API Gateway, Cognito, DynamoDB are the main takeaways from the internship program. Throughout this internship program, I discovered a developer's working life. Besides, the project obliquely assisted me to discover individually, lead a more managed lifestyle and make the mindset to solve problems. This internship program created a way for me to get to know the software engineering industry of Bangladesh.

Bibliography

[1] "monitoring and observability — aws management & governance 2022," 2022.

[2] S. Tilley, *Systems analysis and design*. Cengage Learning, 2019.

[3] R. R. Currie, S. Seaton, and F. Wesley, "Determining stakeholders for feasibility analysis," *Annals of tourism research*, vol. 36, no. 1, pp. 41–63, 2009.