

A Laundry Service System

UIT2211 – SOFTWARE DEVELOPMENT PROJECT

A PROJECT REPORT

Guided by

Dr. Durgadevi Velusamy
Assistant Professor, Department of IT.

Submitted by

- *Saranya S* *3122235002113*
- *Saruleka P* *3122235002114*
- *Sarveash Koushik R V* *3122235002115*
- *Shaam A S* *3122235002116*

**SSN COLLEGE OF ENGINEERING,
KALAVAKKAM**

JUNE 2024

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project titled “A Laundry Service System” is the bonafide work of “Saranya S - 3122235002113, Saruleka P - 312223502114, Sarveash Koushik RV- 3122235002115, Shaam AS - 3122235002116”, and is submitted for project viva-voce examination held on 22.06.2024.

Signature of examiner(s)

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Dr. Durgadevi Velusamy and Dr. Kulandaijosephinejulina J who gave us the opportunity to work on this project. A special thanks to our teachers Dr. Kulandaijosephinejulina J and Dr. Durgadevi Velusamy for their constant guidance and support throughout the project.

Secondly, we would like to thank our parents, relatives and friends who constantly supported and encouraged us to complete this project within the limited time frame.

We are highly obliged to all those who guided and helped us out in completing this project successfully.

TABLE OF CONTENTS

1. INTRODUCTION	5
2. AGILE METHODOLOGY	9
3. PROBLEM STATEMENT	18
4. LITERATURE SURVEY	19
5. REQUIREMENTS	21
6. TECHNOLOGY ASSESSMENT	23
7. DESIGN AND DEVELOPMENT OF SOLUTION	24
8. PROCESS MANAGEMENT	27
9. DEVELOPMENT	28
10. GAP ANALYSIS	34
11.CONCLUSION	38

CLIENT CERTIFICATE

Name of the project: Laundry Service Provider

Name of the Client: Mrs. Sekar

Rating System - 1: Strongly disagree 2: Disagree 3: Neutral 4: Agree 5: Strongly Agree

Questions	1	2	3	4	5
The problem was well discussed and, the requirements and goals were clear.					✓
The project plan was well defined and communicated from the start.					✓
The resources were adequate for achieving the goals.					✓
The original timeline was realistic and was followed.				✓	
The teamwork was well demonstrated.					✓
The client was communicated on regular intervals and given updates on the progress of the project.				✓	
The expected project requirements have been satisfied.					✓

I am very much satisfied by this project.
All my requirements are fulfilled and I
kindly appreciate the team members for
their efforts.

Sekar

Signature of the client.

CHAPTER 1

INTRODUCTION

A small-scale laundry service provider is seeking to enhance its operations and customer service through the development of a sophisticated laundry service management system. The current manual processes for handling customer details, tracking laundry transactions, and generating operational reports are proving to be inefficient and prone to errors. This inadequacy is leading to data inconsistencies, operational delays, and limited insights into business performance. The proposed system aims to address these challenges by implementing an integrated solution that streamlines customer data management, transaction processing, and service point allocation while providing comprehensive reporting capabilities.

1.1 Laundry Service System

The proposed laundry service system will serve as an all-encompassing solution designed to improve the management of customer information, laundry transactions, and service operations. This system will enable the service provider to create, view, and update customer details with ease, ensuring that all customer data is accurate and up-to-date. When customers deposit clothes for laundering, the system will generate a transaction slip with a unique ID, which includes details such as the type of each cloth, the type of laundry service required, associated costs, and the expected delivery date. This will ensure that both the customer and the service provider have a clear record of the transaction.

Internally, the system will automate the allocation of clothes to the appropriate service points based on the type of laundry service required and the availability of resources. This automation will streamline the handling and processing of laundry items, reducing the risk of errors and delays. The system will track the progress of each laundry item, from deposit to collection, ensuring timely delivery to the customer. The transaction will be marked as closed once the customer makes the payment and collects their clothes. Additionally, the system will feature robust reporting capabilities, allowing the service provider to generate daily, weekly, and monthly reports to gain valuable insights into business performance and support strategic decision-making.

1.2 Methodology

The development of the laundry service system will follow an Agile methodology, ensuring that the system is built incrementally and iteratively with continuous feedback from stakeholders. The project will be divided into several phases:

Requirement Analysis: Detailed requirements will be gathered from the service provider, focusing on customer management, transaction processing, service point allocation, and reporting needs.

System Design: A comprehensive design of the system architecture will be created, including database schema, user interface layouts, and backend processes.

Development: The system will be developed using a modern frontend framework (e.g., Tkinter), a robust backend framework (e.g., Python), and a relational database (e.g., Linked list, Priority queue). Development will proceed in sprints, with each sprint focusing on implementing specific functionalities.

Integration and Testing: All system components will be integrated, and thorough testing will be conducted to ensure seamless interaction between the frontend, backend, and database. User acceptance testing will validate the system against the service provider's requirements.

Deployment: The system will be deployed in a live environment, and training will be provided to the service provider's staff to ensure smooth adoption and usage.

Maintenance and Support: Ongoing maintenance and support will be provided to address any issues and incorporate future enhancements based on user feedback.

1.3 Objectives

The primary objectives of the laundry service system are as follows:

Enhanced Customer Data Management: Develop a system that allows easy creation, viewing, updating, and deletion of customer details, ensuring data accuracy and consistency. This functionality ensures that customer information remains accurate and up-to-date, promoting better service delivery and customer satisfaction. Real-time validation mechanisms will be implemented to prevent input errors and maintain data integrity. Secure storage

protocols will safeguard sensitive customer information, adhering to data protection regulations and bolstering trust between the service provider and its client.

Accurate Transaction Processing: A key objective is to streamline transaction processing through the automated generation of unique transaction slips. These slips will comprehensively detail each cloth item deposited, specify the type of laundry service required, outline associated costs, and provide an expected delivery date. By ensuring each transaction slip is uniquely identifiable, the system mitigates the risk of confusion or duplication, enhancing operational efficiency and customer clarity. Moreover, the system will accommodate various laundry service types, including bespoke requests, facilitating tailored service offerings and precise order tracking.

Automated Service Point Allocation: Create an automated process for allocating clothes to appropriate service points, optimizing resource utilization and minimizing errors.

Seamless Transaction Closure: Integrate payment and delivery tracking into the system, enabling smooth transaction closure upon payment and collection of clothes.

Comprehensive Reporting: Provide robust reporting capabilities that generate detailed daily, weekly, and monthly reports, offering valuable insights into business performance and aiding in strategic decision-making. Customizable reporting options will allow stakeholders to tailor reports to specific needs, ensuring relevance and actionable insights that drive strategic growth.

User-Friendly Interface: Ensure the system has an intuitive and user-friendly interface that facilitates easy use by the service provider's staff.

Scalability and Flexibility: Design the system to be scalable and flexible, accommodating future growth and the addition of new features as needed.

1.4 Motivation

The motivation behind developing a comprehensive laundry service system stems from the need to address several critical challenges faced by the small-scale laundry service provider. Currently, the provider relies on manual processes for managing customer details, tracking laundry transactions, and generating operational reports. These manual methods are inefficient, time-consuming, and prone to errors, which can lead to customer dissatisfaction and operational bottlenecks.

Customer Satisfaction: One of the primary motivations is to enhance customer satisfaction by ensuring that customer data is managed accurately and transactions are processed smoothly. An automated system will allow for quick and accurate generation of transaction slips, detailed tracking of laundry items, and timely notifications about laundry status. This will improve the overall customer experience and build trust in the service provider's reliability and professionalism.

Operational Efficiency: The current manual processes for allocating clothes to service points and tracking their progress are not only inefficient but also increase the risk of errors and delays. Automating these processes will streamline operations, reduce manual labor, and ensure optimal use of resources. This will result in faster turnaround times, improved accuracy, and reduced operational costs.

Data Management: Accurate and consistent data management is crucial for the service provider to maintain customer trust and ensure smooth operations. A system that allows for easy creation, viewing, updating, and deletion of customer details will help maintain data integrity and support better decision-making.

Business Insights: The lack of robust reporting capabilities limits the service provider's ability to gain insights into business performance and customer trends. By providing comprehensive daily, weekly, and monthly reports, the new system will enable the service provider to analyze data effectively, identify areas for improvement, and make informed strategic decisions.

Scalability: As the business grows, the volume of data and the complexity of operations will increase. A scalable system that can handle this growth will ensure that the service provider can continue to deliver high-quality service without disruption.

In summary, the motivation for developing the laundry service system is to improve customer satisfaction, enhance operational efficiency, ensure accurate data management, gain valuable business insights, and support future growth. This system will not only address the current challenges but also provide a solid foundation for the service provider to build upon as they expand their business.

CHAPTER 2

AGILE METHODOLOGY

2.1 INTRODUCTION TO AGILE METHODOLOGY

In this section, we expand upon the development process of SociallyConnected, emphasizing the adoption of Agile methodology and the utilization of various tools to facilitate efficient project management, version control, and collaboration. By embracing Agile practices and employing tools such as Jira, GitHub, and Bitbucket, the development team ensured a flexible and iterative approach to software development, enabling faster feedback loops and continuous improvement.

The Agile methodology was chosen as the development approach for SociallyConnected due to its emphasis on collaboration, adaptability, and incremental progress. Agile methodologies promote close collaboration between team members, stakeholders, and end-users throughout the development process. This collaborative approach fosters effective communication, shared understanding, and a strong sense of ownership among team members.

Key roles within the Agile development process were assigned, including the Product Owner and Scrum Master. The Product Owner served as the bridge between the development team and stakeholders, responsible for gathering requirements, defining user stories, and prioritizing features based on user needs and business objectives. The Scrum Master facilitated the Agile process, ensuring that Agile principles and practices were adhered to, removing impediments, and promoting collaboration and continuous improvement within the development team.

2.2 METHODS ADOPTED

Jira as a Project Management Tool:

Jira was utilized as the primary project management tool to plan, track, and manage the development process. It facilitated efficient task management, backlog prioritization, sprint planning, and progress tracking. Jira's user-friendly interface allowed the development team to create and assign tasks, set priorities, and monitor the progress of each sprint. This allowed for better coordination and improved visibility into the project's status.

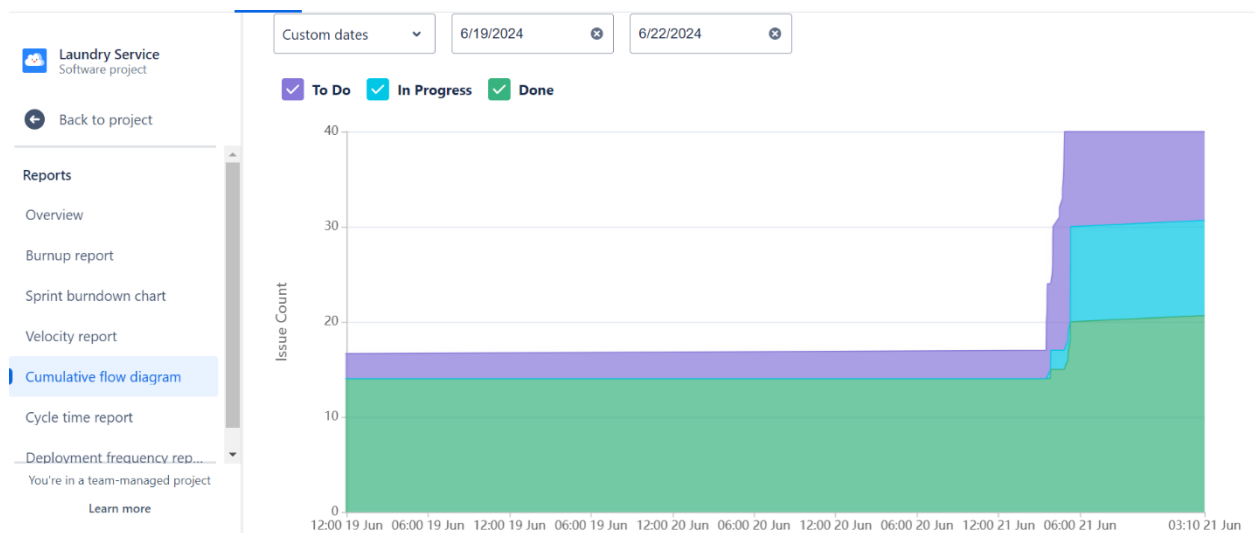


Fig 2.1 Cumulative Flow Diagram



Fig 2.1 Burnt up chart

Product Owner and Scrum Master: The Product Owner and Scrum Master played crucial roles in implementing the Agile methodology. The Product Owner was responsible for defining and prioritizing the product backlog, representing the interests of the stakeholders, and ensuring that the development aligned with the vision and requirements of SociallyConnected. The Scrum Master, on the other hand, acted as a facilitator and coach, ensuring adherence to Agile

principles, facilitating effective communication within the team, and removing any obstacles that hindered progress.

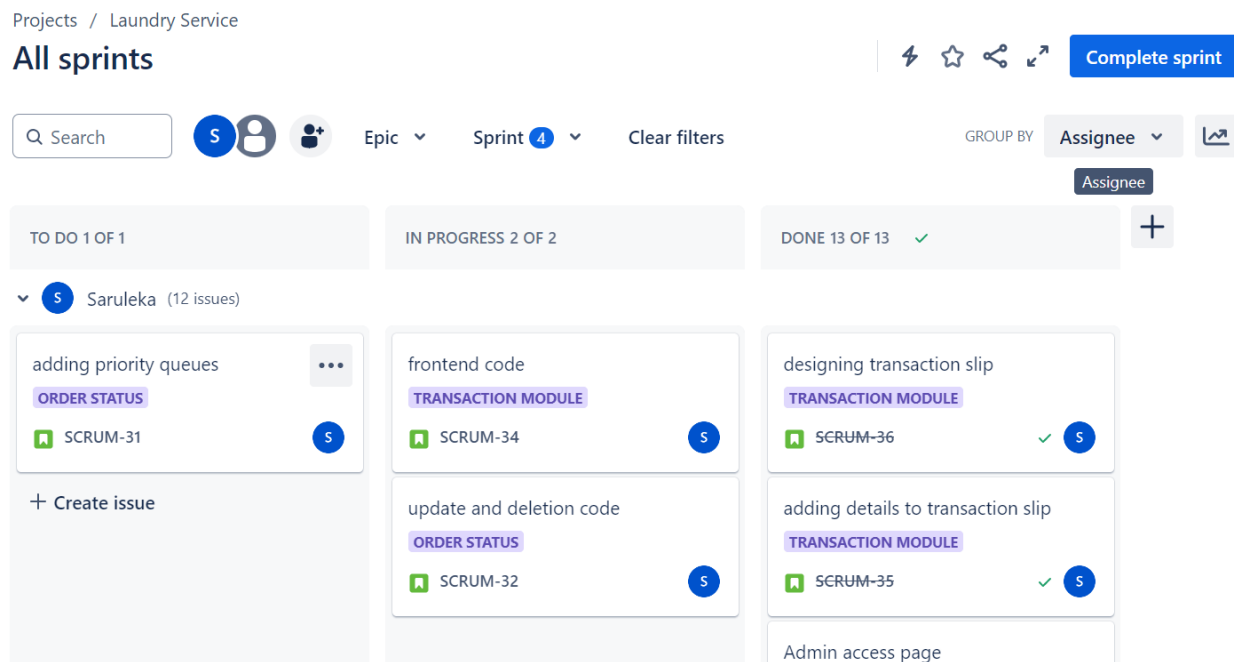


Fig. 2.2 Work status tracking in Jira

Sprint Structure: The development process was divided into sprints, with each sprint having a duration of one week. Sprints are time-boxed iterations that allow the team to focus on delivering a set of features or functionalities within a defined timeframe. The short duration of one week sprints allowed for quicker feedback, frequent retrospectives, and the ability to adapt and respond to changing requirements.

Version Control and Code Repository: GitHub and Bitbucket were utilized as version control and code repository management tools. These tools enabled collaborative development, version tracking, and seamless integration with other development tools. They provided a centralized platform for the development team to work on different features simultaneously, manage code merges, and maintain a complete history of code changes.

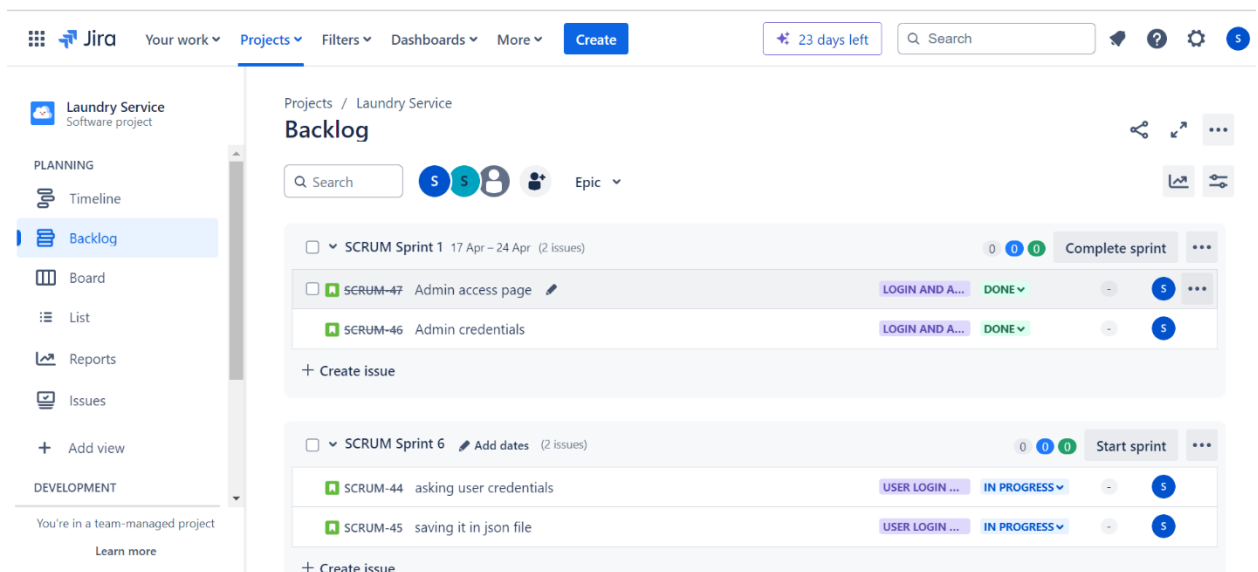


Fig 2.3 Sprint Structure - Admin and User Credentials

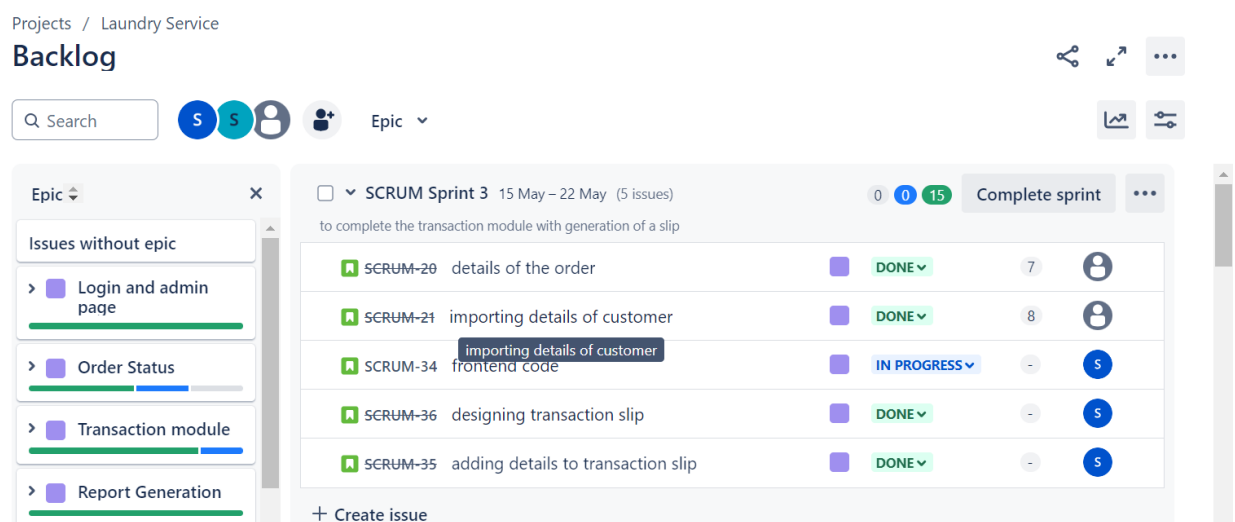


Fig 2.3 Sprint Structure – Transaction details

Backlog

Q Search

S S Epic

2 issues | Estimate: 0

Epic

Issues without epic

- > Login and admin page
- > Order Status
- > Transaction module
- > Report Generation
- > user login and sign up

SCRUM Sprint 2 27 Apr – 18 May (5 issues)

SCRUM-49	Buttons to access view and add customer page	IN PROGRESS	-	S
SCRUM-50	customer credentials for adding customer	IN PROGRESS	-	S
SCRUM-51	viewing the details of customer	IN PROGRESS	-	S
SCRUM-52	updating the details of customer	IN PROGRESS	-	S
SCRUM-53	frontend code design	IN PROGRESS	-	S

+ Create issue

Fig 2.3 Sprint structure – Customer details

Jira Your work Projects Filters Dashboards More Create

23 days left Q Search

Laundry Service Software project

PLANNING

- Timeline
- Backlog
- Board
- List
- Reports
- Issues
- + Add view

DEVELOPMENT

You're in a team-managed project

Projects / Laundry Service

Backlog

Q Search S S Epic

+ Create issue

SCRUM Sprint 5 15 May – 5 Jun (5 issues)

SCRUM-23	importing data from file	REPORT GENE...	DONE	10	
SCRUM-42	monthly report	REPORT GENE...	DONE	-	S
SCRUM-40	daily report	REPORT GENE...	DONE	-	S
SCRUM-41	weekly report	REPORT GENE...	DONE	-	S
SCRUM-39	implementing front end code	REPORT GENE...	DONE	-	S

+ Create issue

Fig 2.3 Sprint structure – Report generation

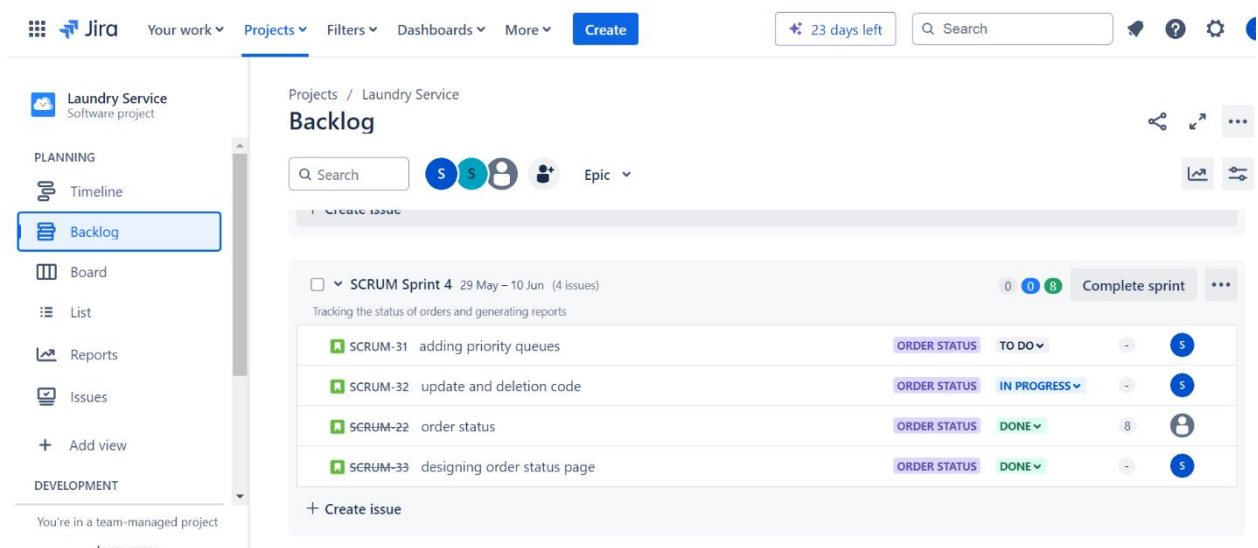
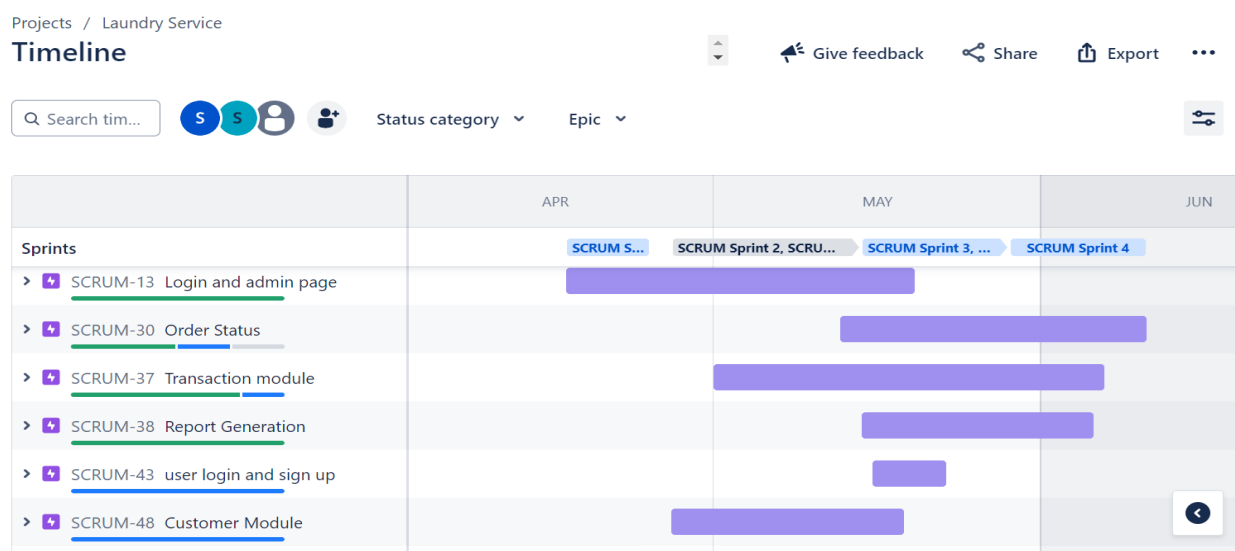


Fig 2.3 Sprint structure – Order status

Continuous Integration and Deployment: The Agile methodology supported continuous integration and deployment practices. By integrating code changes frequently and automating the build and testing processes, the development team ensured a reliable and stable product. Continuous integration practices, supported by tools like GitHub and Bitbucket, facilitated early bug detection and faster resolution, leading to a more robust and high-quality end product.

Iterative Maintenance and Enhancements: After each sprint, the team conducted meetings to reflect on the progress, identify areas for improvement, and plan for the next sprint. This iterative approach allowed the team to incorporate user feedback, make necessary adjustments, and continuously enhance the platform's functionality, usability, and performance.



2.2 Jira Confluence:

Jira and Confluence are two powerful tools that played a significant role in our social media application project. Jira is an issue tracking and project management tool, while Confluence is a collaborative documentation platform. These tools extended beyond the scope of our syllabus, providing essential functionalities and features that greatly contributed to the success of our project.

Confluence:

Confluence played a crucial role in our project's documentation and knowledge sharing efforts. It provided a collaborative platform where team members could create, edit, and share project-related documents, meeting notes, requirements, and design specifications. Confluence's 15 intuitive editor allowed us to structure and organize our documentation effectively.

The screenshot displays a Confluence page titled "2024-05-17 - Meeting notes" within the "Software Development" space. The page header includes navigation links (Home, Recent, Spaces, Teams, More) and a "Create" button. The page content is organized into sections: "Date" (17 May 2024), "Participants" (a list of four users: @saranya2310734, @Saruleka, @shaam2310914, and @sarvaeshkoushik2310680), and "Goals". Below the "Goals" section, there is a "7 days left" badge and a search bar. The left sidebar shows the "Software Development" space with a list of content types (All content, Blogs, Automation, Calendars, Space settings) and a "SHORTCUTS" section. The main content area features a "Action items" section with a checklist of five tasks: "Gather detailed requirements regarding functionality, user roles, and business processes", "Design the system architecture, including database schema, backend, and frontend components.", "Set up the development environment with necessary tools and technologies (e.g., programming languages such as Python)", "Implement the frontend components, ensuring responsiveness and user-friendly design", and "Integrate third-party services if needed (e.g., Mail Notifications)".

Software Development / 2024-05-17 - Meeting notes

2024-05-17 - Meeting notes

Owned by Saruleka, created with a template
Last updated: just a moment ago • 1 min read • 1 person viewed

Date

17 May 2024

Participants

- @saranya2310734
- @Saruleka
- @shaam2310914
- @sarvaeshkoushik2310680

Goals

7 days left

Search

Software Development

All content

Blogs

Automation

Calendars

Space settings

SHORTCUTS

No shortcuts in this space

CONTENT

Software Development / 2024-05-17 - Meeting notes

✓ Action items

- ☐ Gather detailed requirements regarding functionality, user roles, and business processes
- ☐ Design the system architecture, including database schema, backend, and frontend components.
- ☐ Set up the development environment with necessary tools and technologies (e.g., programming languages such as Python)
- ☐ Implement the frontend components, ensuring responsiveness and user-friendly design
- ☐ Integrate third-party services if needed (e.g., Mail Notifications)

Decisions

- To Use Python Tkinter Module to design frontend
- To Use Linked List and Priority Queues as Data Structures
- Separation of Modules within the Group
- Setting deadlines for every task assigned

The screenshot shows a Confluence page titled 'Discussion topics' under the 'Software Development' space. The page contains a table with four columns: Time, Item, Presenter, and Notes. The table lists four discussion topics from a meeting on 2024-05-17. The left sidebar shows the Confluence navigation menu with options like 'All content', 'Blogs', 'Automation', 'Calendars', and 'Space settings'. The top navigation bar includes 'Home', 'Recent', 'Spaces', 'Teams', and a 'Create' button. A search bar and a '7 days left' notification are also visible.

Time	Item	Presenter	Notes
4.00 pm	Project requirements	@saranya2310734	• Clarifying the requirements for developing a laundry service provider system
	Problem Description	@Saruleka	• Understanding the challenges and goals of the project
	Customization of laundry management system	@sarvaeshkoushik2310680	• Discussing the features and options for customizing user accounts
	Data structures and Algorithms	@shaam2310914	• Identifying suitable data structures and algorithms for various functionalities

One of the significant advantages of using both Jira and Confluence was their seamless integration. We could link Jira issues to Confluence pages, allowing for direct access to relevant documentation from within Jira. This integration promoted efficient information flow and contextual collaboration, ensuring that team members had the necessary information at their fingertips while working on specific tasks or issues. Furthermore, the integration enabled us to embed Jira reports, project dashboards, and status updates directly into Confluence pages, providing a centralized view of project progress, metrics, and key performance indicators.

Conclusion:

The incorporation of Jira and Confluence into our social media application project provided us with powerful project management, tracking, documentation, and collaboration capabilities. Jira's robust project management features allowed us to effectively track tasks, manage workflows, and ensure project transparency. Confluence, on the other hand, facilitated collaborative documentation, knowledge sharing, and streamlined communication among team members.

The integration between Jira and Confluence enhanced the overall efficiency of our project by seamlessly connecting project management and documentation. By leveraging these tools, we were able to optimize our project workflows, improve collaboration, and deliver a more streamlined and cohesive project outcome.

By embracing Agile methodologies and utilizing Jira, GitHub, and Bitbucket, the development team of SociallyConnected was able to effectively manage the development process, foster collaboration, and ensure the timely delivery of features. The Agile approach facilitated flexibility, adaptability, and continuous improvement, enabling the team to respond to changing requirements and deliver a high-quality social media platform tailored for college students.

CHAPTER 3

PROBLEM STATEMENT

A small-scale laundry service provider is facing significant challenges in managing customer details and tracking laundry transactions efficiently. The current system lacks the capability to systematically create, view, and update customer information, leading to data inconsistencies and operational inefficiencies. Generating transaction slips with unique IDs and detailed information about each item, such as the type of cloth, type of laundry service required, cost, and delivery date, is cumbersome and inaccurate. The manual process of allocating clothes to appropriate service points is prone to errors, resulting in delays and potential misplacements of items. Additionally, the service provider struggles with tracking the collection and return of clothes, which impacts customer satisfaction and complicates the transaction closure process upon payment and delivery.

Moreover, the service provider is unable to generate comprehensive daily, weekly, and monthly reports, which are crucial for monitoring business performance, understanding customer trends, and making informed decisions. The absence of a robust reporting mechanism limits the ability to analyze data and optimize operations. To address these challenges, there is a need to develop a comprehensive system that can maintain detailed customer records, generate accurate and unique transaction slips, automate the allocation of clothes to service points, and facilitate seamless transaction closure upon payment and delivery. Additionally, the system should support the generation of detailed reports to provide insights into business operations and support strategic planning.

CHAPTER 4

LITERATURE SURVEY

In this literature survey, we explore the existing body of knowledge and research related to the development and implementation of the Laundry Service System. A small scale laundry service provider requires an efficient system to manage customer details, transactions, and reporting. This system needs to handle customer data, generate transaction slips, allocate clothes to service points, and manage payments and delivery. To develop such a system, an examination of similar projects, database design, and algorithms used is essential.

4.1 EXAMINING SIMILAR PROJECTS

a. Laundry Management Systems (LMS)

Several laundry management systems are available commercially, each offering various features. Key systems reviewed include:

CleanCloud is a cloud-based laundry management software that manages orders, customer details, invoicing, and payments. It also provides POS capabilities and reporting tools. GalaWash offers features like order management, customer relationship management, and detailed reporting. It is particularly noted for its ease of use and integration with other software. The Laundry Management System by Cinch focuses on route planning, invoicing, and customer communication. It includes a robust reporting mechanism.

Common features across these systems include customer management, storing and managing customer details such as contact information and transaction history. Order management involves generating unique transaction IDs, detailing each item's laundry requirements, cost estimation, and delivery schedules. Service point allocation assigns tasks to different service points based on laundry type and workload. Payment and delivery tracking record payments, update delivery status, and close transactions. Reporting generates daily, weekly, and monthly reports on transactions, revenue, and service efficiency.

4.2 EXPLORATION ON THE DATABASE

To design a database for the laundry management system, relational databases are often

preferred due to their robustness and ability to handle complex queries. The database schema must capture key entities such as customers, clothes, transactions, service points, and reports.

The customers table stores details like CustomerID, Name, Contact Details, and Address. The clothes table includes fields like ClothID, CustomerID, Type of Cloth, Type of Laundry, Cost, and Status. Transactions are recorded with fields such as TransactionID, CustomerID, Date, TotalCost, PaymentStatus, and DeliveryDate. Reports are generated with fields like ReportID, Date, Type (daily/weekly/monthly), and Content.

The entity-relationship diagram (ERD) illustrates the relationships between these entities. Customers have a one-to-many relationship with transactions, meaning a single customer can have multiple transactions. Transactions have a one-to-many relationship with clothes, indicating that a single transaction can include multiple items of clothing. Service points manage clothes, with each service point capable of handling multiple items.

4.3 ALGORITHM ADOPTED

The system relies on various algorithms for efficient operations.

Unique transaction ID generation can be achieved using a combination of the timestamp and customer ID to ensure uniqueness. For example, a transaction ID can be generated by hashing the combination of CustomerID and Timestamp. Cost calculation is dynamic, based on cloth type and laundry type using predefined rates. The cost can be calculated with an algorithm that multiplies the ClothTypeRate by the LaundryTypeRate.

Service point allocation involves an algorithm that balances the workload across service points. A simple load-balancing algorithm or a more sophisticated weighted round-robin approach can be used to distribute tasks efficiently among service points. Payment and delivery tracking utilize state management algorithms to ensure accurate tracking of payment and delivery statuses. These state transitions, such as pending to paid to delivered, can be managed through a finite state machine.

For reporting, aggregation algorithms summarize transaction data for reports. Jason or dedicated reporting tools can be used to generate daily, weekly, and monthly reports, providing insights into transaction volumes, revenue, and service efficiency.

CHAPTER 5

REQUIREMENTS

Sprint #	Epic	User story#	Requirement	Remarks on implementation
1	SM-1	2 Design of suitable Data Structure	Design a suitable Data Structure to store clothes lists.	Implement the Data Structure to store clothes lists.
1	SM-1	6 Design of suitable Data Structure	Design a suitable Data Structure to generate report	Give suitable Data inputs after integrate the other functions
1	SM-1	4 Design of suitable Data Structure	Give suitable Data inputs	Inputs given for the Two Implemented DataStructure and tested for errors and bugs
1	SM-1	3 Design of suitable Data Structure	Integrate code with other basic function	Other functions (sign-up,log-in) integrated with theimplemented Solution
1	SM-1	9 Design of suitable Data Structure	Give suitable Data inputs after integrate the other functions	Inputs including Log-in credentials given and tested for bugs and errors.
2	SM-10	7 Design of suitable Data Structure	Integrate the complete implementation of Data Structure with all other functions	Above implemented solution integrated with follow/unfollow,view posts,generate Ads,view profile

2	SM-10	11 Implementation of Login page	Username should be validated (Register page)	Checked the given Username satisfies the constraints for username for new user.
2	SM-10	12 Implementation of Login page	Password should be validated (Register Page)	Checked the given Password satisfies the constraints for password for new user.
2	SM-10	13 Implementation of Login page	Username should be checked(Log-in page)	Checked the username given by the user is valid.
2	SM-10	14 Implementation of Login page	Password should be checked(Log-in page)	Check the Password given by existing User is valid for given username
2	SM-10	Implementation of Login page	Integrate frontend and backend.	The developed Back-end to be integrated with the Front-end
2	SM-10	14 Implementation of Login page	Password should be checked(Log-in page)	Check the Password given by existing User is valid for given username
3	SM-15	19 Designing of UI	Button to view customer details	admin will view customer details.
3	SM-15	19 Designing of UI	Button to view customer details	admin will view customer details.
3	SM-15	18 Designing of UI	Separate buttons for the functions implemented in Back-end.	The user have buttons for login, adding clothes and payments

CHAPTER 6

TECHNOLOGY ASSESSMENT

Python:

Python is a high-level, interpreted programming language known for its readability, simplicity, and versatility.

Tkinter:

Tkinter is the standard GUI (Graphical User Interface) library for Python. It provides a powerful object-oriented interface to the Tk GUI toolkit, which is available on most Unix platforms as well as on Windows and macOS.

Graph-Based Friend Sorting Algorithms:

- a. Graph-based algorithms offer an effective approach to sorting and displaying friends and contacts in a social network.
- b. These algorithms enable the identification of meaningful connections and relationships among users, facilitating intuitive navigation and interaction within the platform.
- c. By employing graph-based algorithms, SociallyConnected enhances the user experience by providing a comprehensive understanding of social connections.

Randomized Algorithm for Ad Generation:

- d. The use of a randomized algorithm for generating ads introduces variety and unpredictability, enhancing the advertising experience for users.
- e. Randomized algorithms can reduce bias in ad selection, ensuring a fair distribution of advertisements to users.
- f. By incorporating a randomized algorithm, SociallyConnected aims to increase user engagement with ads and improve the overall effectiveness of advertising campaigns.

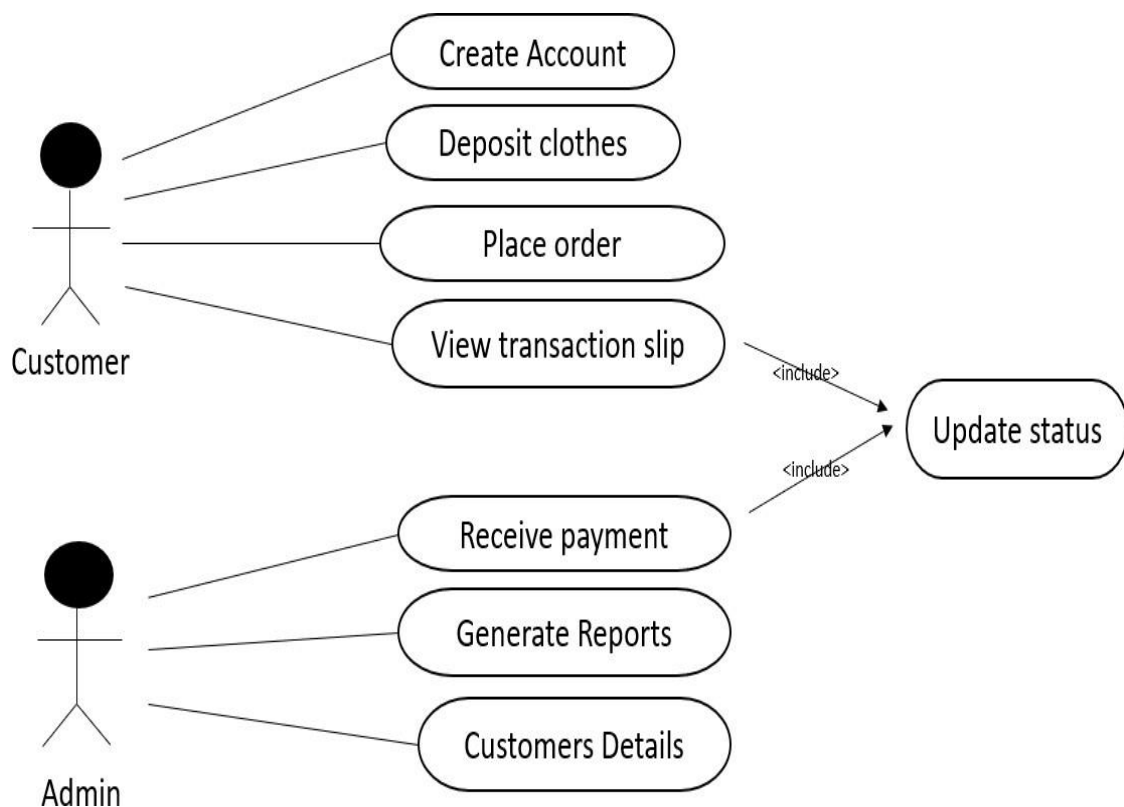
Overall, the technologies, algorithms, and languages used in SociallyConnected are carefully chosen to address specific requirements and challenges. They enable the creation of a dynamic and visually appealing user interface, ensure efficient data management and processing, provide powerful backend support, and enhance the social and advertising experiences of users. The selection of these technologies and algorithms is driven by their proven effectiveness, wide industry adoption, and their ability to meet the unique needs of SociallyConnected as a cutting-edge social media platform for college students.

CHAPTER 7

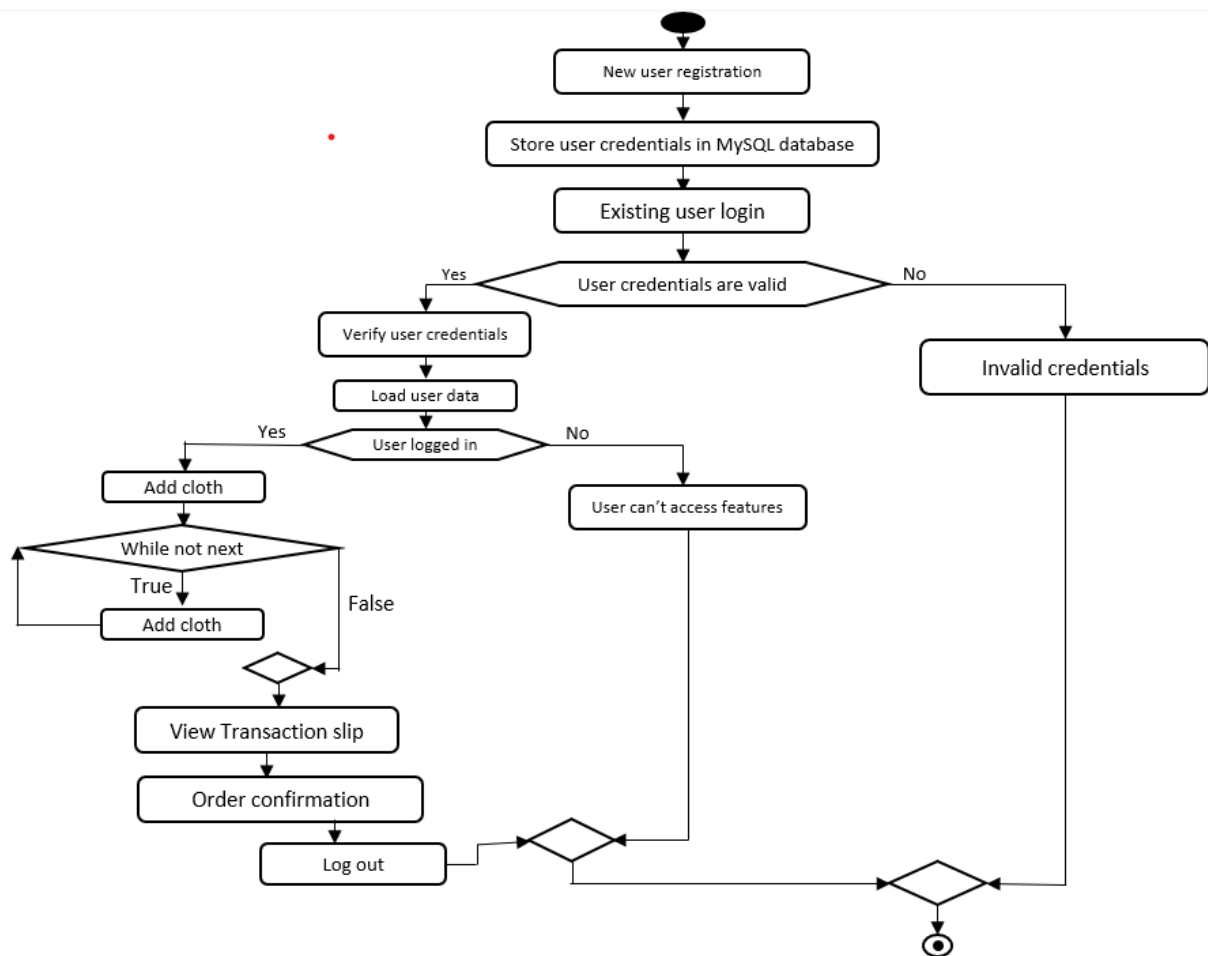
DESIGN AND DEVELOPMENT OF SOLUTION

This chapter discusses the proposed methodology through various use case diagrams and finally describes the system design and customer side application workflow.

7.1 USE CASE DIAGRAM



7.2 WORKFLOW



7.3 RISK MANAGEMENT

Risk Area	Risk Description	Mitigation Strategies
User Interface (UI) and User Experience (UX)	Poor usability or confusing interface design could lead to user dissatisfaction.	Conduct user testing and gather feedback early. Follow UI/UX best practices and guidelines.
Security	Vulnerabilities (e.g., SQL injection, XSS) could lead to data breaches or unauthorized access.	Implement input validation and sanitization. Use secure storage methods. Implement authentication and authorization mechanisms.
File Handling	Errors in file handling (e.g., missing files) could disrupt functionality or lead to data loss.	Implement robust error handling and validation. Ensure appropriate permissions and access controls.
Exception Handling	Unhandled exceptions could result in application crashes or unexpected behavior.	Use try-except blocks for error handling. Implement logging for error tracking and debugging.
Data Integrity	Data corruption or inconsistency due to concurrent access or improper manipulation.	Use transactions or locking mechanisms for data consistency. Validate and sanitize input data.
Performance	Poor performance with large datasets or complex operations could lead to user dissatisfaction.	Optimize code for efficiency. Consider asynchronous programming for long-running tasks.
Dependency Management	Dependency version conflicts could cause instability or security vulnerabilities.	Regularly update dependencies. Use virtual environments or containerization for dependency isolation.
Scalability and Maintenance	Difficulty in scaling the application or maintaining codebase as it grows.	Design modular and reusable code. Follow design patterns (e.g., SOLID). Plan for scalability needs.
Testing and QA	Insufficient testing coverage could lead to undiscovered bugs or regressions.	Implement automated testing (unit tests, integration tests). Perform thorough testing across environments and edge cases.
Documentation	Inadequate documentation could hinder onboarding and troubleshooting efforts.	Maintain up-to-date documentation (architecture, API, troubleshooting guides). Encourage knowledge sharing within the team.

CHAPTER 8

PROCESS MANAGEMENT

LAUNDRY SERVICE SYSTEM

Process management is crucial for developing an efficient system for a small-scale laundry service provider. This system aims to manage customer details, generate transaction slips, allocate clothes for laundering, handle payments, and generate reports. Here's how we approach process management for this project:

1.Project Planning: Before starting development, we will define the project objectives and requirements. We'll create a detailed project plan with milestones such as system design, implementation phases, testing, and deployment.

2.Agile Methodology: We will adopt Agile methodology to manage our development process effectively. This will allow us to break down tasks into manageable user stories such as managing customer details, generating transaction slips, handling payments, and generating reports. We will work in iterations to deliver incremental functionality.

3.Version Control: Git will be used for version control to manage our codebase. This will enable multiple developers to collaborate on the project simultaneously, track changes, and ensure code integrity.

4.Development Process: We will develop the system using Python and Tkinter for the frontend. Data structures such as dictionaries, tuples, lists, and queues will be utilized for managing customer details, storing transaction information, and handling clothing allocation. The system will be modular, with components for customer management, transaction handling, and reporting. Each team member will be assigned specific modules to work on concurrently.

5.Code Reviews: Regular code reviews will be conducted to maintain code quality. Team members will review each other's code, provide feedback, and ensure consistency and correctness in implementation.

6.Testing and Bug Tracking: Thorough testing will be performed to validate system functionality and performance. Test cases will cover scenarios such as customer data management, transaction processing, and report generation. Bugs and issues will be tracked using Jira, and necessary fixes will be implemented.

7.Documentation and Knowledge Transfer: Comprehensive documentation will be prepared, covering system architecture, design decisions, Tkinter UI components, data structure implementations (like dictionaries for customer details, queues for transaction processing), and user guides. Knowledge

transfer sessions will be conducted among team members to share insights and ensure smooth collaboration.

8.Reporting : The system will generate daily, weekly, and monthly reports as per the service provider's requirements. These reports will include transaction summaries, revenue insights, and customer activity details.

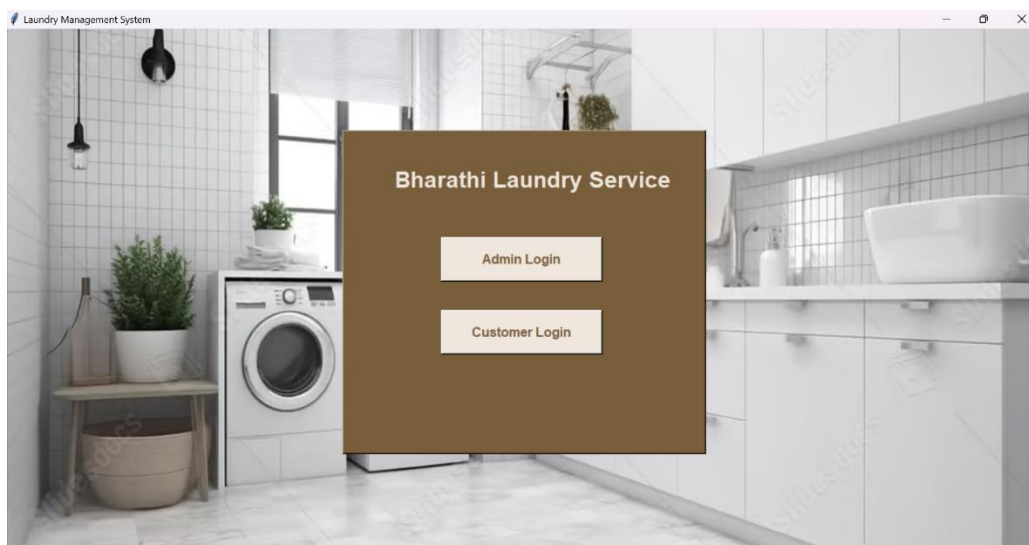
By implementing effective process management techniques tailored to the laundry service provider's needs, we aim to deliver a reliable and efficient system that meets their operational requirements and enhances customer service.

CHAPTER 9

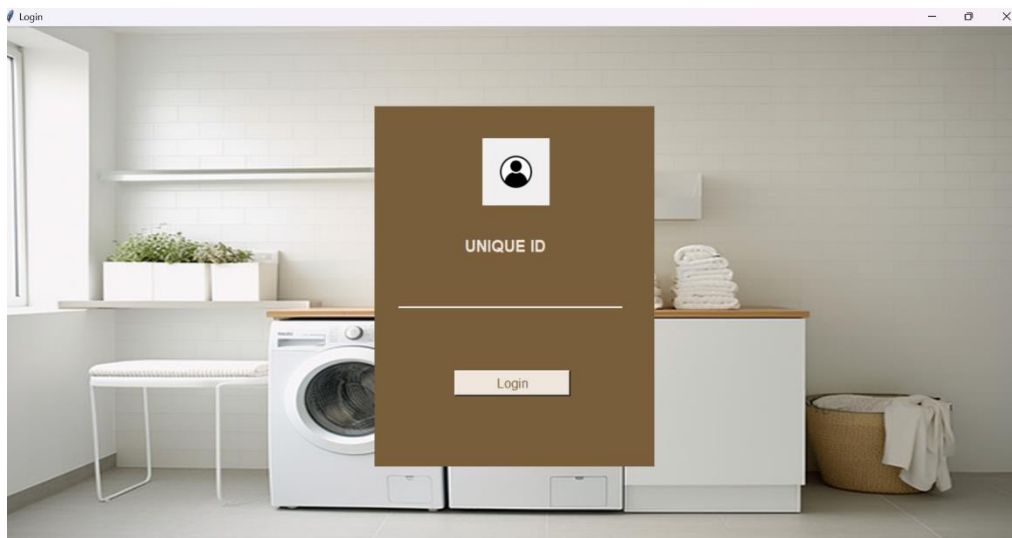
DEPLOYMENT AIM AND OBJECTIVES

This chapter provides an overview of a project aimed at developing a system for a small-scale laundry service provider to manage customer details efficiently. The system's objectives include creating, viewing, and updating customer profiles, as well as generating unique transaction slips upon clothes deposit. These slips will detail cloth types, laundry requirements, costs, and delivery dates. The system will internally allocate clothes to appropriate service points and manage the retrieval process for each customer. Transactions will be finalized upon customer payment and clothes collection, with the system capable of generating daily, weekly, and monthly reports for operational insights.

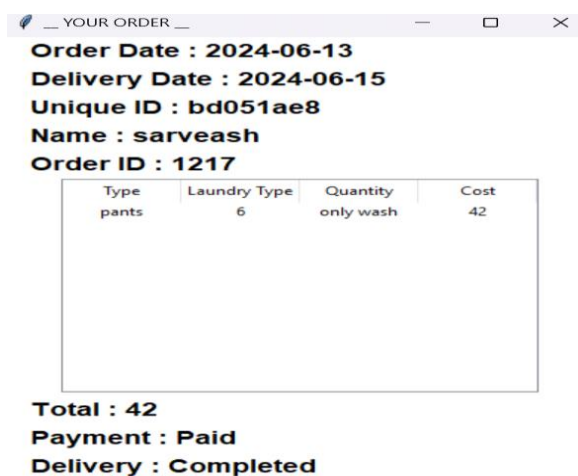
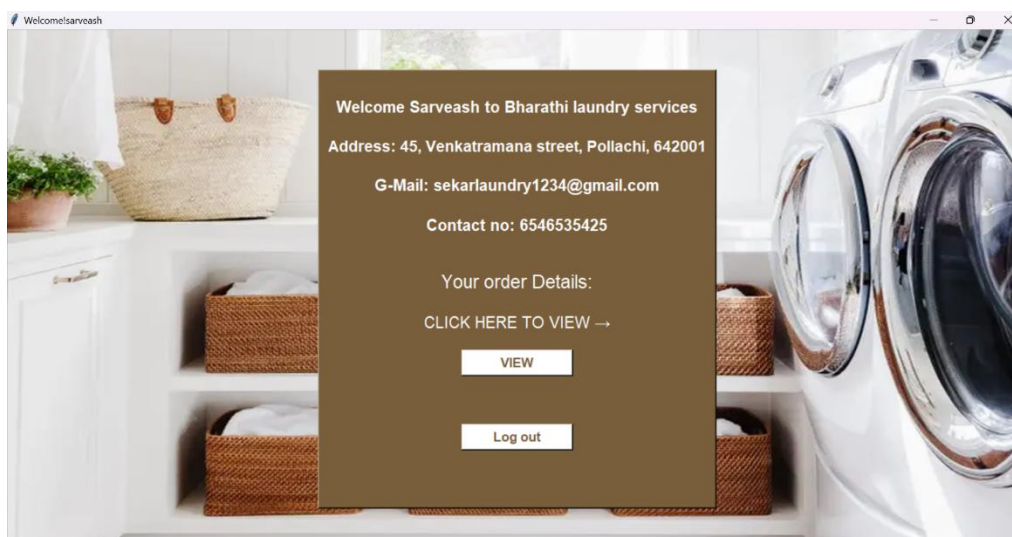
WELCOME PAGE



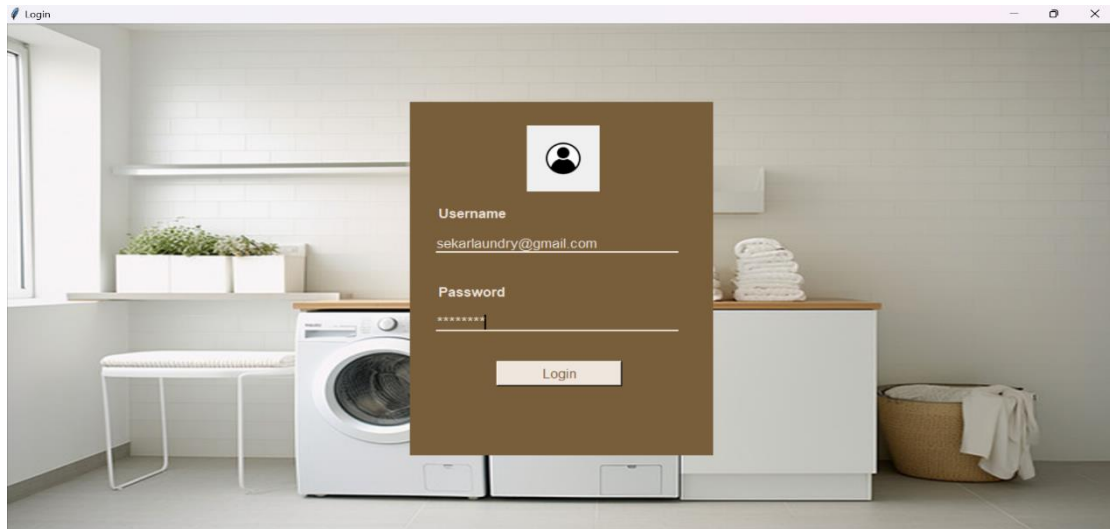
CUSTOMER LOG IN PAGE



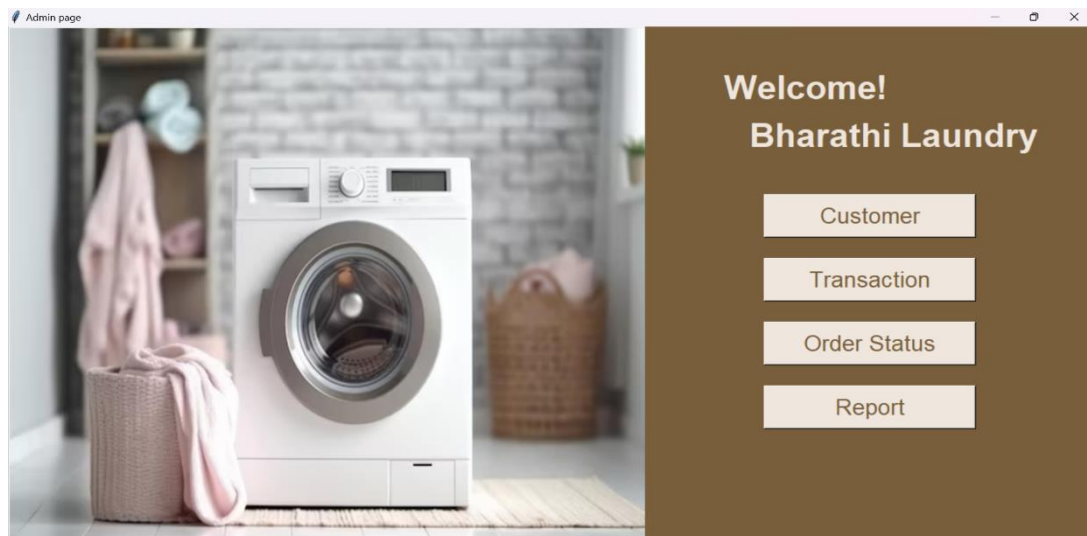
ORDER DETAILS



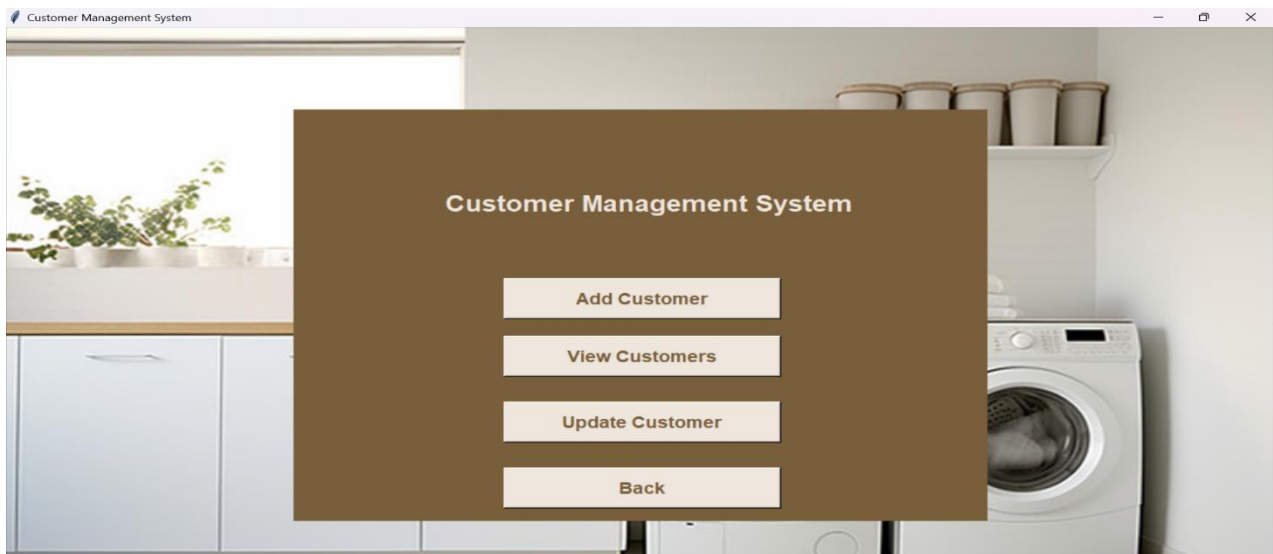
ADMIN LOG IN PAGE



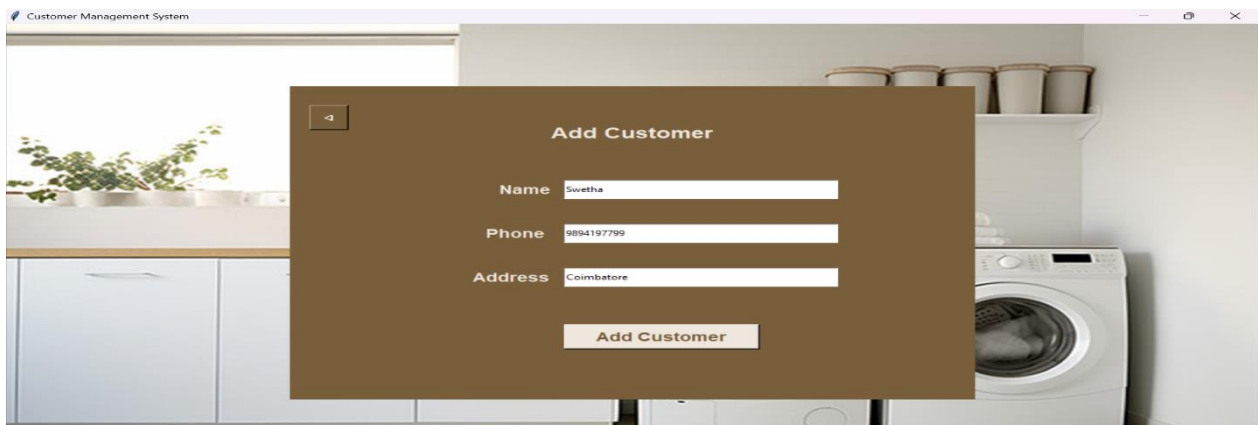
AFTER LOG IN



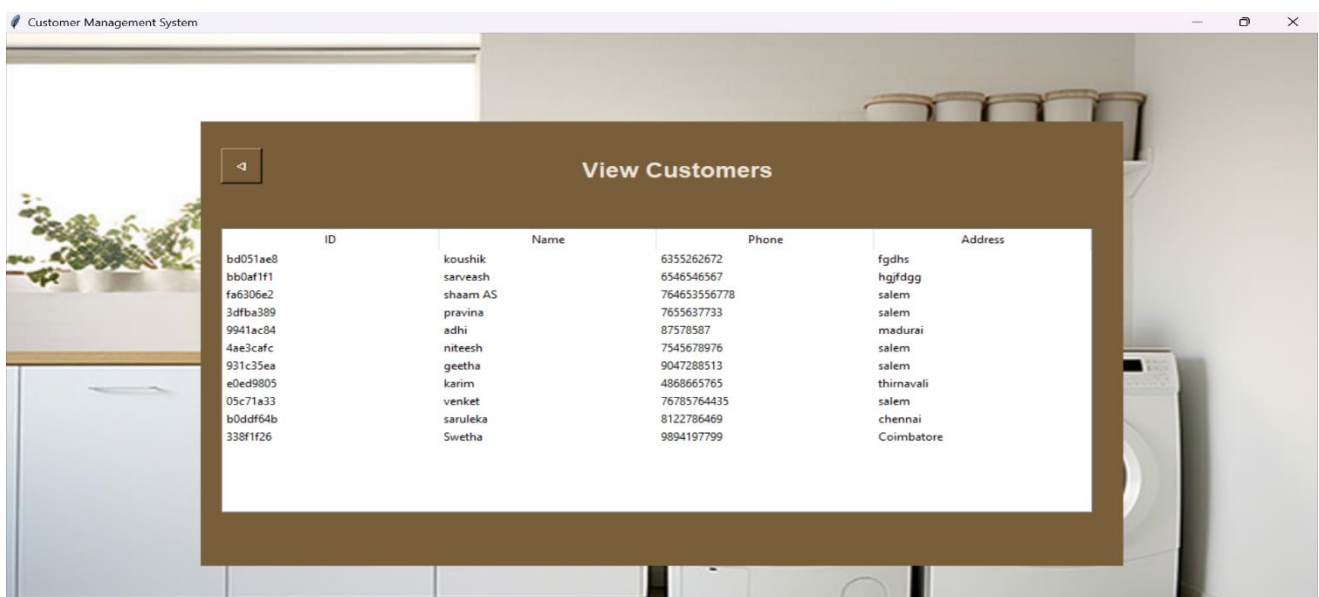
CUSTOMER MANAGEMENT



ADD CUSTOMER



TOTAL CUSTOMERS



UPDATE CUSTOMER

Customer Management System

◀

Update Customer

Name Swetha

New Phone 9715872655

New Address chennai

Update Customer

TRANSACTION MANAGEMENT

__TRANSACTION__

Transaction Slip

UNIQUE ID : ad139b3b

NAME : Shaam

S.No	Type	Laundry Type	Quantity	Cost
1	Shirt	dry wash	4	28
2	Pant	wash and dry	5	50

S.No	Type	Laundry Type	Quantity
3	T-shirt	<div>only wash dry wash iron wash and dry wash,dry and iron</div>	

add

NEXT

__TRANSACTION Slip__

Bharathi laundry services

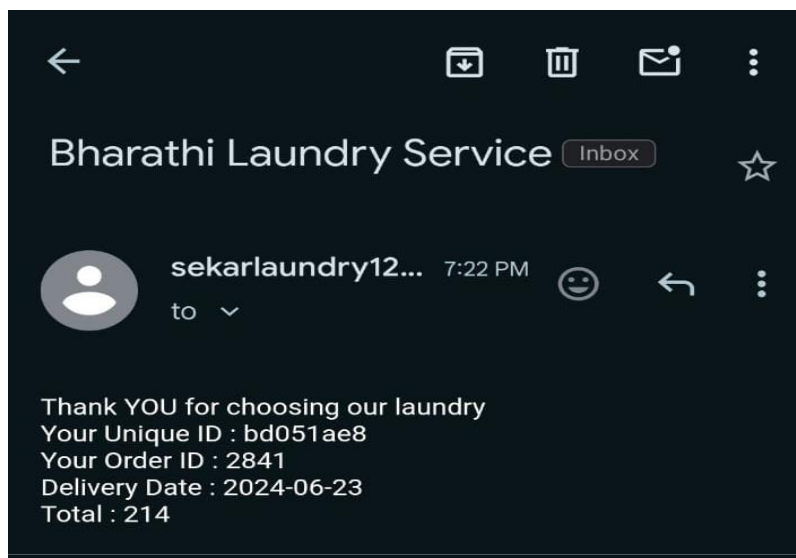
Address : 45,Venkatramana street,pollachi,642001
G-Mail : sekarlaundry@gmail.com
Contact no : 6546535425

Client Details :

Name : Shaam Date : 2024-06-14
Unique ID : ad139b3b
Phone no: 9342708253
G-mail: Shaam@gmail.com
order No : 6187
Total Items : 11
Total : ₹ 92
Delivery Date : ₹ 2024-06-16

NEXT

MAIL NOTIFICATION



ORDER STATUS

Order Status

Order ID: Delivery Status:

Payment Status:

Order Date	Order ID	Customer Name	Payment Status	Delivery Status
2024-06-13	0339	koushik	Paid	Completed
2024-06-13	9772	adhi	Paid	Pending
2024-06-14	2276		pending	pending

REPORT MANAGEMENT

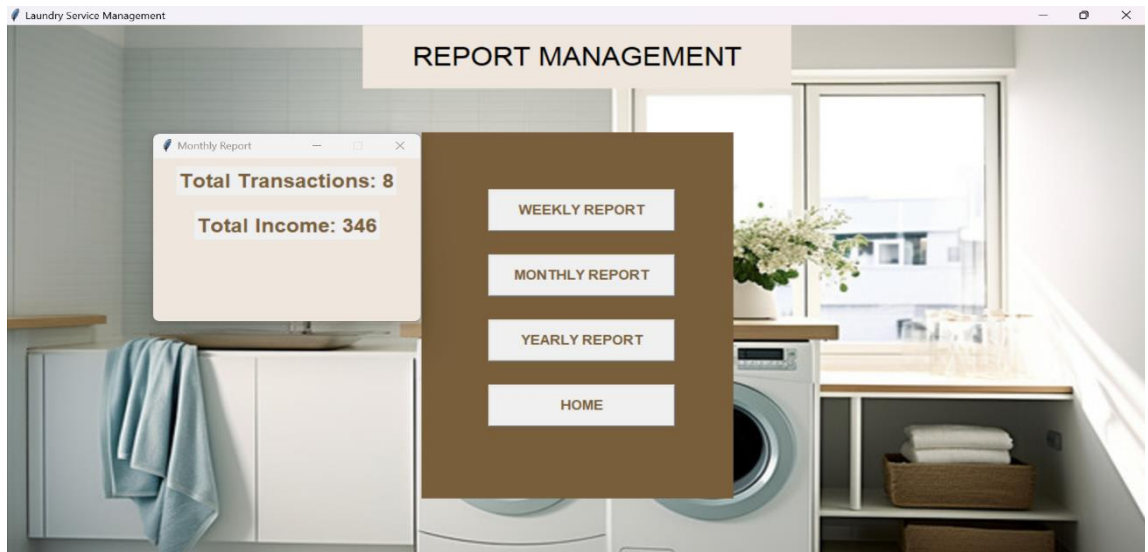
Laundry Service Management

REPORT MANAGEMENT

WEEKLY REPORT



MONTHLY REPORT



YEARLY REPORT



CHAPTER 10

GAP ANALYSIS

10.1 Introduction:

The development of our project involved exploring various tools, technologies, and concepts that were beyond the scope of our syllabus. These additions greatly enhanced the functionality, user experience, and overall effectiveness of our application. In this gap analysis, we will identify and evaluate the gaps that emerged due to the introduction of these new elements.

We will analyze the impact of these gaps on different aspects of the project, including project management, database management, data serialization, frontend integration, frontend design tools, graph algorithms, and graph databases. Additionally, we will discuss the mitigation strategies we employed to overcome these gaps, which involved extensive research, self-study, collaboration, and seeking external guidance.

By conducting this detailed gap analysis, we aim to provide insights into the challenges we faced, the measures we took to address them, and the overall impact on the project's success. This analysis will showcase our ability to adapt, learn, and incorporate new tools and technologies, highlighting our commitment to delivering a high-quality social media application that meets the evolving demands of users in today's digital landscape.

10.2 Detailed Gap Analysis for Laundry Service System Project

10.2.1 Project Management - Jira:

- **Gap:** The project scope did not include the usage of project management tools like Jira, resulting in a lack of familiarity and expertise among team members.
- **Impact:** This gap had a significant impact on our project management processes. Without Jira, we lacked a centralized platform for task tracking, assigning responsibilities, and monitoring progress. It resulted in challenges in maintaining an organized workflow, tracking individual contributions, and identifying potential bottlenecks or delays.
- **Mitigation:** To bridge this gap, we conducted an in-depth exploration of Jira's features and functionalities. We dedicated time to self-learning and utilized online tutorials and resources to understand Jira's capabilities better. Additionally, we sought guidance from external

consultants and project management professionals to gain insights into best practices and efficient usage of Jira. Through continuous training and hands-on practice, we successfully incorporated Jira into our project management workflow, we created roadmaps, epics and sprints, enabling better project planning, task allocation and progress tracking, improving overall efficiency and collaboration.

10.2.2 Desired Future State

Frontend (Alternative Framework)

- Develop an enhanced, user-friendly interface using a modern frontend framework like React or Angular. This would provide a more responsive and intuitive user experience.
- Implement comprehensive forms for viewing, updating, and deleting customer details, supporting complete CRUD operations with real-time feedback and validation.
- Create detailed transaction slip forms that allow dynamic data entry for various types of clothes, specific laundry services required, associated costs, and expected delivery dates.

Backend (Alternative Technology)

- Develop an advanced mechanism for generating unique transaction IDs to ensure consistency and uniqueness across all transactions.
- Introduce an automated, efficient system for allocating clothes to appropriate service points, ensuring optimal resource utilization and minimizing errors.
- Fully integrate payment and delivery status tracking into the transaction lifecycle, enabling seamless transaction closure and improved customer satisfaction.

Database (Alternative Solution)

- Design a well-structured and normalized database schema with clear relationships between tables to support efficient data management and retrieval.
- Develop optimized queries and stored procedures to enhance performance and ensure quick response times, even as data volume grows.
- Implement stringent constraints and validation rules at the database level to ensure data

integrity and consistency

Reporting (Enhanced Features)

- Develop robust reporting capabilities using tools like Tableau or Power BI, allowing the generation of detailed daily, weekly, and monthly reports.
- Introduce customizable reporting options based on various criteria such as customer, service point, and transaction status.
- Enable export functionality for reports in multiple formats (e.g., CSV, PDF) to facilitate data analysis and sharing.

10.2.3 Gap Analysis

Frontend (Tkinter to Modern Framework)

- The current Tkinter forms are functional but lack advanced features and user-friendliness. Transitioning to a modern frontend framework like React or Angular will provide a more responsive and intuitive user experience.
- Current forms do not fully support CRUD operations. Implementing comprehensive forms with real-time feedback and error handling in a modern framework will bridge this gap.
- Existing transaction slip forms offer limited data entry. Developing detailed forms with dynamic fields in a modern framework will allow for comprehensive data entry and improved user interaction.

Backend (Python to Robust Framework)

- The current backend uses basic CRUD operations with limited validation. Moving to a robust backend framework like Django or Flask will optimize these operations and enhance validation.
- The mechanism for generating unique transaction IDs is simplistic. An advanced mechanism in a robust framework will ensure uniqueness and consistency.
- Current allocation logic for service points is inefficient. An automated system in a robust framework will improve resource utilization and minimize errors.

- Minimal integration of payment and delivery tracking. Fully integrating these aspects into the transaction lifecycle in a robust framework will ensure seamless transaction closure and customer satisfaction.

Database

- The current database schema is basic and needs refinement. Transitioning to with a well-structured and normalized schema will support efficient data management.
- Existing database queries are not optimized. Developing optimized queries and stored procedures will enhance performance and ensure quick response times.
- Limited data integrity and validation mechanisms are in place. Implementing stringent constraints and validation rules will ensure data integrity and consistency.

Reporting (Basic to Enhanced Features)

- Reporting capabilities are currently limited. Developing robust reporting features using tools like Tableau or Power BI will provide comprehensive insights and facilitate data analysis.
- Customization options for reports are minimal. Introducing customizable reporting options and export functionality will meet diverse reporting needs and improve usability

By identifying these gaps and taking proactive measures to address them, such as conducting self-study, collaborating with external resources, seeking guidance from experts, and engaging in continuous learning, we successfully incorporated new tools, technologies, and concepts into our social media application project. These efforts enabled us to overcome challenges and deliver a feature-rich, robust, and scalable application that meets the requirements of modern social media platforms.

CHAPTER 11

CONCLUSION

11.1 Challenges

Data Migration: Moving existing data to a new, normalized database structure can be complex and may involve significant data transformation and validation efforts.

User Training: Training staff to use the new, more advanced system interfaces and functionalities will be necessary to ensure a smooth transition and minimize disruption to operations.

Integration and Testing: Ensuring that all components (frontend, backend, database) work seamlessly together will require rigorous integration testing and iterative improvements.

11.2 Future Scope

Enhanced Customer Experience: By leveraging modern frameworks and technologies, the system can provide a more intuitive and responsive user interface, improving customer satisfaction and operational efficiency.

Scalability and Performance: The new system architecture will be more scalable and capable of handling increased data volumes and user interactions, supporting future growth.

Advanced Reporting and Analytics: With robust reporting tools, the service provider can gain deeper insights into operations, enabling data-driven decision-making and strategic planning.

Automated Workflows: Automation of allocation, tracking, and reporting processes will reduce manual efforts, minimize errors, and increase overall productivity.

Expanded Features: Future enhancements could include mobile app integration, customer notifications via SMS/email, and advanced payment gateways, further improving service offerings and customer engagement.

