Intro to SWE 3313: Automation of J's

Developed by: *Shams Hasan, Ryan Tran, Isa Siddique, Prince Duepa, Nhut Tran*

Table of Contents:

Scope

Goal: Introduce automation to help restaurants with analyzing day-to-day and by-the-hour tasks such as:

1. Coordination of Work Activities
    - The system will utilize algorithmic methods to address relevant tasks for employees and customers alike.

2. Anticipating and Handling Periods of Low/High Patron Traffic:
    - The system will incorporate features to predict and manage variations in patron traffic. This involves utilizing historical data and real-time analytics to forecast peak hours and anticipate slow periods.
    - By understanding and preparing for fluctuations in customer demand, the restaurant can optimize staffing levels, inventory management, and overall resource utilization.

3. Recognizing Trends Early Enough:
    - The system will enable the early identification of trends in customer preferences, allowing the restaurant to adapt quickly to changing market demands.
    - By analyzing data on menu item popularity and sales trends, the restaurant can make informed decisions about promoting bestsellers or discontinuing less popular items, ultimately maximizing revenue.

4. Lowering Operating Costs, and Increasing Efficiency/Productivity and Profits:
    - Automation will contribute to lowering operating costs by reducing manual errors, minimizing wastage, and optimizing resource allocation.
    - Increased efficiency and productivity will result from streamlined processes, faster order processing, and improved communication.
    - The overarching goal is to enhance overall profitability by creating a more agile and data-driven operation that can respond proactively to market dynamics.

<u>Support Employee Roles:</u> Host, Waiter, Cook, Busboy, & Manager

<u>Direct Links:</u>

- Host -> Waiter
- Waiter -> Cook
- Busboy -> Host

<u>Privileges for each Employee Role:</u>

- Each role has their personalized home screen & refreshes automatically

    - I.e. table is marked ready, table's order is prepared, the host assigns a waiter to a table, etc
- Manager has admin powers over all roles
    - create/modify profiles, track employee activity, & authorize restricted waiter activities
    - Managers can manage inventory, tracking, & sale analysis
- Waiters
    - Greeted with floor status screen (assigned tables are colored in)
    - Coloring: green = open, yellow = occupied, & red = dirty

Functionalities

1. Order Management System
    - Staff can quickly input customer orders through UI
    - Access to real-time order status updates for kitchen staff & wait staff
    - Implement electronic order delivery to kitchen (minimizes delays)
2. Table Management
    - Host can view/manage the status of all tables with UI
    - Host can assign servers to specific tables
    - Real-time updates on whether the table is open, occupied, or dirty
3. Staff Coordination
    - The Cook can indicate to the waiter when a meal is ready to be served
    - The Waiter can notify the busboy to clean the table (links back to table management)
    - The Busboy can notify the host when the table is ready for customers
4. Predictive Analytics for Patron Traffic
    - Uses past business days to predict peak hours/slow periods
    - Provide recommendations for adjusting staffing levels based on traffic
5. Menu Management
    - Digital menu with detailed description, prices, and pictures
    - Implement analytics for popular items and find trends
6. Financial/Operation Reports
    - Generate daily/hourly reports on revenue and revenue percentage per menu item'
    - Analyze average turnaround times and prep times for orders
7. User Auth & Access Control
    - Have role-based access control to restrict certain functionality based on employee role
8. Data Backup
    - Implement regular data backups to prevent data loss in case of system failures.

Schedule

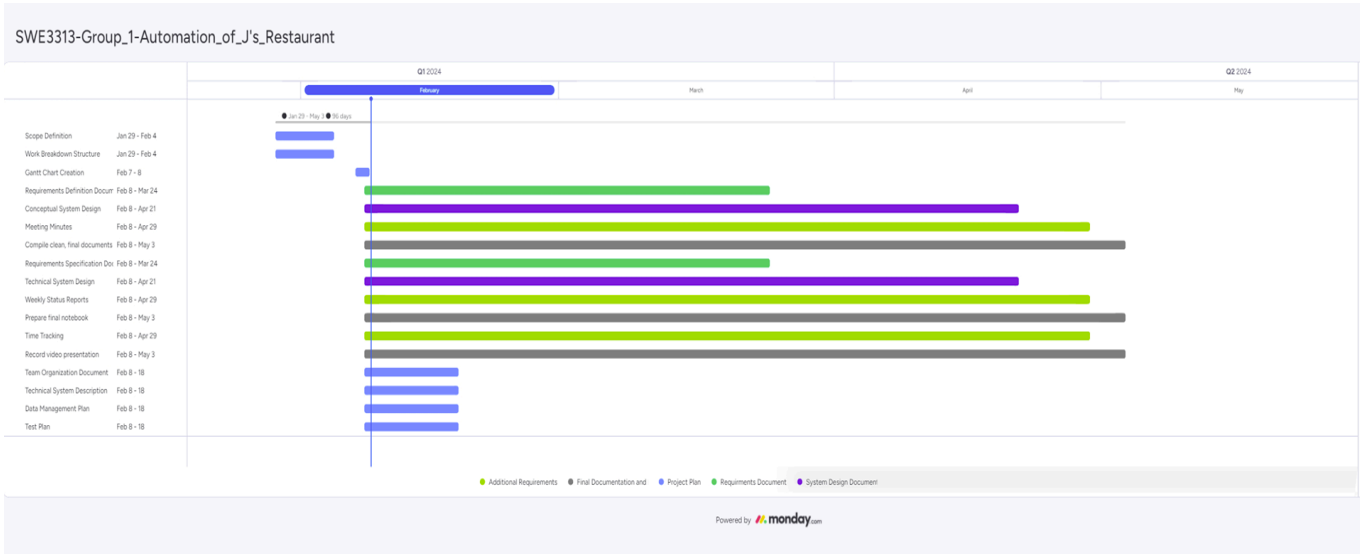
Task ID	Work Breakdown Structure	Planned Start	Planned Finish	Workload-planned	Workload-actual	Progress
1	• Create 5 Java Classes corresponding to employee roles • Host Class: created method to view and manage the status of all tables & assigning servers (Hashmap & List) • Waiter Class: implemented methods headers for order management & real-time updates with Data Structures • Cook Class: created method header for when meal is ready with Data Structures • Busboy Class: implemented method header to notify when table is ready and added a Data Structure to establish cleanliness • Manager Class: created admin powers and managed inventory.	2/4/24	2/5/24	building the basic functionality of classes and simulating data using in-memory data structures.	Built the basic functionality of classes and simulating data using in-memory data structures.	100 %
1_1	• Goal: Finish Planning Document for Submission	2/5/24	2/17/24	• Add Resumes & Integrate into Github Repo • Gantt Chart • Technical Summary • Data Management Plan • Test Plan	• 5/5 Resume(s) Added • Gantt Chart Added • Test Plan, Technical Summary, and Data Management plan complete.	100 %

1_2	• We are going to need 8 Database Tables 1) Employee Table: Store information about each employee, including their roles and authentication details. 2) Table Status Table: Keep track of the status of each table in the restaurant. 3) Order Table: Store information about customer orders. 4) OrderItem Table: Track individual items within each order. 5) MenuItem Table: Store details about each menu item. 6) StaffCoordination Table: Log interactions and coordination between staff members. 7) AnalyticsData Table: Store historical data for analytics purposes. 8) Inventory Table: Keep track of inventory details for menu items.	2/19/24	2/24/24	Create database (mySQL) for Menu items & other planned databases		0%
2	• ER diagram creation: • Needs to show its data and relationships in the database • Should have appropriate design patterns	2/4/24	2/24/24	• Using 1+ pages for the ER diagrams • Avoid using tables		0%

2_1	• Complete the 1st prototype of the project • Needs to have proper visuals and design, along with adequate drawings and must match finalized requirements	2/26/24	3/10/24	• Creating enough screens in order to properly show the representation of final product • Creating a seamless UI experience		0%
2_2	• Complete the Requirements Documents • This document should specify all functional and non-functional requirements. • It should also specify all use cases, with a diagram and flow of events, and have an ER Diagram. • There should also be class diagrams and documentation, as well as a decision table.	3/11/24	3/24/24	• Creating 2 documents: a requirements document, and a requirements specification document.		0%
3	• Complete the System Design Documents. • This document should have a conceptual system design and a technical design. • It should also describe the database tables and technical support specifications.	3/25/24	4/21/24	• Create/upload relevant screenshots • Create samples from the system such as restaurant receipts, and report summary • Create relevant ER and UML diagrams		0%
3_1	• Complete the second product prototype. • Specifically complete the waiter components of the system.	4/22/24	5/3/24	• Creating proper source code. • Creating an executable jars		0%

3_2	• Submit final project/presentation • The Second Prototype should be the final product. • Final Documentation must also be completed	5/3/24	5/3/24	• Complete the final notebook • Complete team evaluations		0%

Gantt Chart



Link to the Gantt Chart in our Github Repository:

[SWE3313-Automation-Project/Project Plan/SWE3313-Group_1-Automation_of_J's_Restaurant.pdf at main · RyanTren/SWE3313-Automation-Project (github.com)]

Roles & Resumes

Shams Hasan: Project Lead and Co-Developer
Ryan Tran: Co-Developer and Co-Project Lead
Prince Duepa: Main Software Developer
Isa Siddique: UI Architect
Nhut Tran: Database Developer and Reporting Analytics

Shams Hasan

678-451-6377| 1shamshasan@proton.me | www.linkedin.com/in/shams-n-hasan/ | https://github.com/shamshasan0/

Education

Kennesaw State University — **Kennesaw, GA**
Bachelor of Science in Software Engineering — *Expected Graduation: Fall 2025*
Cumulative GPA 3.5

Experience

Computer Science Tutor — October 2023 – December 2023
Georgia State University — *Atlanta, GA*

- Skillfully directed students through the intricacies of object-oriented programming (OOP), data structures, and algorithm design
- Empowered students to construct and implement robust classes, seamlessly integrating methods and objects to solve real-world problems
- Cultivated a mastery of variable naming conventions, including camel-casing, to enhance code readability and maintainability
- Demonstrated exceptional communication and interpersonal skills, fostering a supportive and encouraging learning environment

Projects

MILKY WAY WONDERS [NASA] | *HTML, CSS, JavaScript, JSON, REST*

- Developed a dynamic web application using the fetch API to retrieve live NASA images and videos from the NASA API
- Implemented JSON parsing techniques to extract and interpret data from the NASA API responses.
- Utilized DOM manipulation to dynamically display live NASA images and videos along with informative explanations on the website
- Employed media queries to ensure responsive design and optimal viewing across various devices, including mobile phones, desktops, laptops, and tablets
- Successfully introduced the app to over 40 individuals, captivating their interest in the wonders and intricate nature of the galaxy

MidEast Foodies | *HTML, SASS, TypeScript, Angular*

- Collaborated with an experienced front-end developer to design and implement a user-friendly and visually appealing website showcasing a collection of authentic Middle Eastern recipes
- Implemented a user-friendly search and filtering system to allow users to easily find recipes based on their dietary preferences, cooking time, and meal type

Book-io Scraper | *HTML, XML, Python*

- Developed a robust web scraping application using Python to extract book titles from a popular bookstore website based on their star ratings
- Optimized the scraping process to efficiently scan through hundreds of pages of books, significantly reducing processing time and improving resource utilization
- Implemented a structured data extraction approach, categorizing scraped book titles into five distinct star rating categories (1-star, 2-star, 3-star, 4-star, and 5-star)
- Successfully utilized lxml, BeautifulSoup, and Requests libraries to seamlessly parse HTML content and extract relevant book title data from the target website

Technical Skills

Languages: HTML, CSS, JavaScript, TypeScript, Java, Python, R, Angular, SQL
Tools and Technologies: AWS, Git, Shell, WordPress, Microsoft Office, Eclipse, IntelliJ, PyCharm, VS Code
Operating Systems: Mac, Windows, Linux

Ryan N. Tran
CS Undergraduate | concepting@protonmail.com | (678)-670-9868 | LinkedIn: ryantren | GitHub: RyanTren

SKILLS

Programming: Java, C#, C++, Swift (iOS), SwiftUI, Storyboard, Objective-C, Nuke
Tools: IntelliJ, PyCharm, Eclipse, XCode, Git

EXPERIENCE

CodePath — **San Francisco, California**
IOS Development Student — *Oct 2023 – Dec 2023*

- Built applications from scratch using Apple's native IDE (Xcode) and Swift programming language.
- Successfully developed a personal custom app as part of the semester project.
- Created engaging user interfaces (UIs) with logic and cloud-based APIs, using libraries (Nuke), demonstrating creativity and problem-solving skills.

DreamChasers — **Atlanta, Georgia**
Sales Representative — *Aug 2022 – Present*

- Collaborated within team settings to strategize and construct vending stands within prominent stadiums, including Mercedes-Benz Stadium and Truist Park.
- Proficient in operating Point of Sale (POS) systems to facilitate smooth and efficient transactions for Fanatics, Inc.

PROJECTS

Motional (Exercise App) — **Atlanta, Georgia**
Creator/Developer — *Oct 2023 – Dec 2023*

- Developed Motional, an iOS fitness app, with Swift as the primary language for core logic & user-friendly interfaces/features.
- API integration using Alamofire, seamlessly connecting the app to the Exercises API by API-Ninjas to provide users with a comprehensive list of targeted exercises.
- Optimized user experience through the implementation of the Nuke library for efficient image loading and display, enhancing the visual appeal of the app.

EDUCATION

Kennesaw State University — **Atlanta, Georgia**
B.S. in Computer Science — *Expected Graduation, May 2026*

- **Concentrations:** Artificial Intelligence and Data Analytics
- **GPA:** 3.72/4.00, *Dean's List*
- **Related Coursework:** Data Structures & Algorithms, Object-Oriented Programming, Data Communication, Discrete Math, Telecommunications,
- **Activities:** KSU Vietnamese Student Association

Isa Siddique

US Citizen | isiddique713@gmail.com | (470)-535-4481 | LinkedIn: LinkedIn | GitHub: isasiddique

EDUCATION

Kennesaw State University **Marietta, Georgia**
B.S. in Software Engineering *Expected Graduation, May 2025*

- **GPA:** 3.98/4.00
- **Related Coursework:** Database Systems, Object-Oriented Programming, Technical Writing.

EXPERIENCE

SMART Center at Kennesaw State University **Atlanta, Georgia**
Student Tutor *Aug 2023 – Present*

- Assist students in college-level Math courses such as College Algebra, Precalculus, Calculus, and Statistics.

Kumon North America, Inc. **Marietta, Georgia**
Tutor *Jun 2022 – May 2023*

- Graded Homework and Classwork assignments whilst answering students' questions.

PROJECTS

Language Translation Website **Marietta, Georgia**
Individual *Aug 2023 – Dec 2023*

- Developed a front-end application using HTML, SQL, and PHP for language translation.
- Hosted the website on a local XAMPP server to connect the SQL and user input.

Company/Product Pitch **Marietta, Georgia**
Team Secretary *Oct 2023 – Dec 2023*

- Led a team of 3 to pitch and design a fictional company and product.
- Wrote technical documentation such as a pitch, memo, team charter, and progress report.

ACTIVITIES AND LEADERSHIP

Muslim Student Association **Marietta, Georgia**
Member *Jan 2022 – Current*

- Member of an organization of 490+ members, hosting religious events, game nights, and fundraisers.

SKILLS

Programming: C#, Java, SQL, HTML, PHP
Tools: Visual Studio, Notepad++
Personal: Mandarin Fluency, Technical Communication, Analytical Thinking, Time Management

PRINCE DUEPA

(208) 949 – 8913 ▪ kannahprince12@gmail.com

TECHNICAL SKILLS

Software Engineering

- Adept in Python, Java, and shell scripting
- Proficient with Docker
- ETL design and principals with Apache Airflow
- Proficient in various development environments including Linux, macOS, and Windows
- Git, Make, Gradle, and various software development life-cycle tools
- Familiar with Scrum roles and methodology
- REST API design
- Software testing principals and designs; Experience with JUnit and Pytest testing framework

IT Support/Administration

- Proficient troubleshooting various technology issues
- Adept with all flavors of Windows, macOS, and Linux

Web Development

- Skilled in JavaScript, CSS, and HTML
- Adept with Python's Flask
- Understanding of HTTP/S and TCP/IP

Databases

- Relational databases experience with Microsoft SQL Server and PostgreSQL

PROFESSIONAL EXPERIENCE

Application Analyst Specialist August 2018 – present ▪ Athens, Georgia
University of Georgia

- Provide technical support for SAGE, the university's student advising tool
  1. Troubleshoot and resolve all issues reported by advisors, students and other stake holders
  2. Work with Office of Instruction - Academic Advising Services to make the tool more advisor friendly
  3. Work with vendor to enable and pilot new features
- Develop and maintain SAGE Management, an in house application use to manage student connections
  1. Python powered REST API back-end using Flask and Microsoft SQL Server
  2. React front-end using React-Admin framework
  3. Provide technical support and bug fixes as well as enhancements
  4. Engage in monthly users' group meetings to acquire feedback
- Develop and maintain SAGE ETL
  1. ETL process written in Python for Apache Airflow that gathers data from various campus sources to populate SAGE
- Implement Bitbucket DevOps CI

1. Run all tests and provide code coverage reporting
2. Build application
3. Create Dockerfile and docker-compose for all projects

- SME on Apache Airflow, an open-source workflow authoring and monitoring tool. In addition to SAGE ETL, also worked on

1. Workflow to update UGA's L-Soft LISTSERV membership
2. Workflow to update Active Directory Group membership using results of MS-SQL Server stored procedures

IT Service Desk Technician January 2018 – August 2018 ▪ Boise, Idaho

FlexTechs

- Open, manage, and maintain service tickets using Cherwell

Undergraduate Research Assistant June 2017 – December 2017 ▪ Boise, Idaho

Boise State University

- Most of my time was spent developing DockoMatic, a open-source, desktop based protein docking automation application. My involvement included feature development, refactoring, and collaborating with a team of faculty.
- Researched and implemented methods to automate the installation of DockoMatic on AWS.
- Researched and implemented methods to surface job progress and error message to front-end GUI.
- Developed, and documented a Bash installer for all flavors of Linux.
- Transition the project from Java to Electron using HTML, JavaScript and CSS using Electron.

Engineering Intern January 2016 – August 2016 ▪ Boise, Idaho

Z-Data, Inc.

- I lead work on Greenplum bench-marking tool on IaaS providers AWS, Azure, and Google Cloud Platform.
- I worked extensively with HashiCorp open-source DevOps and infrastructure tools Terraform and Vagrant.
- Participated in weekly SCRUM stand-ups.
- Used Chef to automate DevOps tasks.

Tutor June 2015 – January 2016 ▪ Boise, Idaho

Boise State University

- Tutored for CS-121 (Java I) and CS-221 (Java II). During open lab I helped students with varying programming experience with their questions. Assisted students in weekly hands-on lab.
- Reinforce good programming habits as well as OOP principals and structures.

Senior IT Support Assistant February 2013 – May 2015 ▪ Boise, Idaho

Boise State University

- I handled over 200 support tickets, fully resolving and documenting issues for future reference.
- Managed and deploy OS images as well as software packages using SCCM 2012 for the College of Business and Economics.
- Managed and deploy printers, network drives, and user accounts with Active Directory Domain Services.
- Use HP Web JetAdmin and Embedded Web Server to manage a fleet of 22 printers. I served as lead on the HP Beta and Delta test programs at the College of Business and Economics.
- I provided support for both on-site and remote clients that included students, staff and faculty.
- Setup and tear-down of various technologies and software.

NHUT TRAN

Email: ntran47@students.kennesaw.edu
LinkedIn: https://www.linkedin.com/in/nhut-tran-207807288
GitHub: https://github.com/ntt256

EDUCATION

Kennesaw State University (Honors Program)

Bachelor of Science: Computer Science

Expected Graduation Date: May 2025

Technical Skills: Java, Swift, Python

EXPERIENCE

CodePath

iOS Development (September '23 – Nov '23)

- Worked in groups/independently using XCode platform on MacOS to build custom iOS apps and UI design
- Learned how to implement API into coding projects using PostMan application

Technical Summary

UI:

Using the Java Spring Boot framework, we can create login screens for different employees (servers, other kitchen staff, and managers) for them able to access their tasks and duties within their job role. Spring Boot is a tool that allows us to create web applications and/or microservices; it is a way for us to build web apps in a fast way with all of these several login screens. For example, the manager's screen will have the most pages (selections) since they are the admin of the restaurant. They will be able to see all the ongoing activities within the restaurant such as employees during that shift, which tables are open, etc. Note that upon opening the app the screen will be universal for all restaurant employees since the device has not been logged in yet.

The customer will have the ability to order menu items and analyze the details of the menu item such as the price and any deals. The menu can also be modified by the restaurant employees to change menu pricing, and remove, or add items to the menu. There will also be data collected to see what page or part of the menu/UI the customers use or browse the most. The menu will consist of multiple pages, separating each category by type (entrees, appetizers, drinks, etc) to keep it organized. There will be a final page for confirming the order of items made by the customer(s).

Connecting the backend to a database:

To access all of this information at our disposal, we can use a JDBC Connector to connect our Java Hibernate backend to our MySQL database, enabling us to grab information from our tables using CRUD commands that act similarly to REST. JDBC is part of Oracle, which is a database management system, that uses API to allow the user to query information, such as information from relational databases. Our MySQL database will hold records for different classes (classes being the employee position: Manager, Server, etc.) and the columns and rows will contain data relevant to their position and what the permissions they are enabled to: make (CREATE), analyze (READ), change (UPDATE), and clear/get rid of (DELETE).

With Maven, we will go into our projects pom.xml file and connect our JavaFX application by creating a dependency. After this we will create our MySQL connection with a Database tool plug-in integrated within IntelliJ in the applications.properties file.

Spring Boot connection w/ mySQL Database example:
spring.datasource.url=jdbc:mysql://localhost:3306/(your_database)

spring.datasource.username=(your_username)
spring.datasource.password=(your_password)
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update

Then we will create entity classes in java that will persist our database (Application Java Class using Spring Boot)

Design:

To design our executable application, we will utilize front-end language, CSS, with JavaFX to create a visually appealing Java GUI that will take on a responsive state with desktops/laptops. Additionally, the design for the login screen will look aesthetically simple for all employees. Each employee will have a different home screen once they login, for example the Manager will see admin permissions and tables and every other employee will be only able to see certain things.

Hardware Requirements:

Our web-app will support devices such as smartphones, tablets, and desktops. The processing power and memory of the devices need to be sufficient enough to be supported by our system. Touchscreens will be required for smartphones and desktops for our web-app service to function properly. Most smartphones will be compatible with our system in terms of storage (at least 64GB).
For desktops and laptops, a minimum of 4GB RAM is required as it is not very heavy on the device. 4GB is standard for web browning and simple application usages. The minimum required RAM should not be a problem since most laptops and desktops today carry 8GB RAM. A processor (CPU) of at least Intel i3 is required for the functionality of the web-app or an AMD Ryzen 3 is the minimum required if using desktops/laptops that carry that type of processor. The minimum CPU required for the web-app service ensures that the user experience is simple and straightforward.

Software Requirements:

Having all of our services and repositories from Spring framework, we will use IntelliJ to set up models in order to grab necessary data from our mySQL database. Using Maven will allow our developers to manage our project swiftly and build dependencies quickly. Spring Boot, mySQL, and JavaFX (GUI tool) will be implemented to create the foundations of our software.

- Maven: Project management tool for Java projects, allows you to manage project's build lifecycle, dependencies, and documentation.

    - Build Lifecycle: Maven defines a set of standard build phases (e.g., compile, test, package, install, deploy) that a project goes through during

its build process. Each phase represents a different step in the project's lifecycle.

- Project Object Model (POM): Maven uses a Project Object Model, defined in an XML file called pom.xml, to configure and manage a project. The pom.xml file includes information about the project's dependencies, plugins, build settings, and other configuration details.
- Dependency Management: Maven simplifies the management of project dependencies. You can specify dependencies in the pom.xml file, and Maven will automatically download and manage the required libraries from a central repository or other specified repositories.
- Plugins: Maven relies on plugins to extend its functionality. There are numerous plugins available for tasks like compiling code, running tests, generating documentation, and more. Developers can configure these plugins in the pom.xml file.
- Repository: Maven uses a repository to store and retrieve project artifacts (JARs, WARs, etc.) and their dependencies. There are central repositories provided by the Maven community, and you can also set up your own repository.
- Convention over Configuration: Maven follows the principle of convention over configuration, which means that it uses default configurations based on conventions, reducing the need for extensive configuration files. However, developers can customize configurations when needed.

- Spring Boot: Framework used to build Java-based applications such as this standalone project. It is designed to simplify the development of Java applications by providing a set of conventions, defaults, and features that streamline the development process and make it easier to create robust, production-ready applications.

    - *Why Spring Boot?*
        - The software we are creating is a standalone software, meaning we don't need a web connection (the internet) to run our software and this is because Spring Boot works with our POM (Maven) to get libraries offline.
        - Using Spring Boot with our Java GUI application streamlines development, simplifies configuration, and provides a standardized way to build, package, and distribute our application. It helps us focus on building features and functionality rather than spending time on intricate setup and configuration details.
        - In terms of Backend, Springboot lets you integrate RESTful API's for the frontend to communicate with the backend seamlessly from client to server.

    - With databases you can integrate Spring Data JPA with Spring Boot to interact with a database to retrieve menu items, customer orders, and etc.
    - If we were to create our software without spring, we would have to do more configuration and planning with creating project structure, GUI, RESTful API, Database connection, and Dependency Management.

- mySQL Workbench: mySQL is a relational database management system that allows developers to store data in separate tables for optimized speeds. With this database system we can use data tables, views, rows, and columns to store data. It's almost like MS Excel, for a simpler explanation. For our software, mySQL will be used to store employee info (usernames/passwords), menu items for J's Restaurant, and analytics. (determining what items on the menu are performing the best & etc. etc.)

Data Management Plan

(formatted this based on IBM's guide on making a data management plan:
What is data management plan? | IBM)

Purpose

The purpose of our Data Management Plan for the Automation of J's Restaurant is to establish the guidelines for the efficient and secure management of data within our system. Our plan aims to achieve accuracy, integrity, and accessibility of data for J's Restaurant personnel while maintaining privacy and complying with relevant data protection regulations. Having these guidelines will allow: efficient restaurant operations, user friendly staff interaction, and valuable analytics for managers to assess.

Data Definitions

Customer Data: Information related to customer profiles, including contact details and order history.

Order Data: Details of customer orders, including items, quantities, and timestamps.

Staff Data: Information about restaurant staff, including roles, responsibilities, and access privileges.

Table Data: Status and information related to restaurant tables, including occupancy, cleanliness, and assigned wait staff.

Menu Data: Details about menu items, categories, and pricing.

Managerial Data: Information pertaining to managerial activities, such as revenue metrics, personnel efficiency, and sales analysis.

Data Collection/Access

Customer Access: Customers can access and update their profiles, view order history, and modify delivery addresses and payment methods.
Waiter Access: Waiters can view floor status, select tables, and add items to the table's tab. They have access to real-time updates on orders and can interact with the kitchen staff.

Manager Access: Managers have comprehensive access to revenue metrics, personnel efficiency, floor status, and menu management. They can create and modify staff profiles, track employee activities, and authorize restricted waiter activities. Managers can also manage other aspects of restaurant operations, such as inventory tracking and sales analysis.

FAQs

1) *How can customers modify their delivery addresses?*

   Customers can log in and update their profiles, including delivery addresses, through the user interface.

2) *What information can waiters view on the floor status screen?*

   Waiters can view assigned tables, with colors indicating their status (green for open, yellow for occupied, red for dirty).

3) *How does the manager access revenue metrics?*

   The manager can log in and access detailed breakdowns of revenue, sales percentages per menu item, and personnel efficiency.

Research Data Limitations

The data management plan prioritizes security, recognizing that while robust measures are implemented to safeguard customer and operational data, the system is not entirely foolproof. Continuous monitoring and updates are integral to promptly address emerging security threats. The effectiveness of the system is contingent upon user proficiency, and to ensure optimal utilization, comprehensive training and support resources will be provided, mitigating potential user-related issues. Additionally, strict adherence to data protection regulations is imperative, and the system will undergo updates as needed to align with evolving legal requirements, ensuring continuous regulatory compliance.

By addressing these components, the data management plan aims to provide a comprehensive framework for handling data within the Restaurant Management System, ensuring the system's efficiency, security, and compliance with relevant standards and regulations.

Test Plan

Creating high quality, reliable and secure applications requires extensive testing. As such we aim to have several layers of testing as we develop the project.

Unit Testing:

Unit tests are important as they allow us to test individual methods in isolation to make sure its implementation is correct. This can help us uncover bugs early on. Unit tests will be incorporated throughout the development process. These can be automated using a testing framework to run tests quickly and frequently. By systematically executing the software and comparing the actual results against expected results, testers can discover issues that may have occurred during the development phase.

Functional:

As each component/feature of the app is completed a functional test should be performed. That is, check if the feature is implemented according to requirements and does the expected input yield the desired output. This testing verifies that the software functions according to the defined requirements and specifications. It ensures that all features and functionalities work as intended, providing a reliable user experience.

Integration:

For the integration of different parts of the systems, it is important that each component of the app work together with minor issues. For example, it is crucial that when we implement API's into the system, it is performing properly by communicating and fetching the requested data. It is also important for our back-end and the front-end components to interact with each other properly and effortlessly.

Performance:

Performance testing will collect the results/data of the application's speed and responsiveness. It is also important to test that our system's overall performance will still be in good shape no matter the amount of incoming traffic or data sent into it. Evaluating the application's performance comes in different ways such as testing it with light work and the heavy work. It is important to see how our system responds to different types of workload to ensure our system's health is in good shape for overall performance.

Security:

Security testing helps identify vulnerabilities and weaknesses in the software that could be exploited by malicious entities. This is crucial for protecting sensitive customer data and the integrity of our application. Key aspect of security testing for the application are:

- Verify authentication system to only authorized user can access the application
- Additionally, test the role based access control to know that users only access features that's assigned to their role
- Perform input validation on user provided inputs.
- Check how the application guard against SQL injection, cross-site scripting (XSS) and other type of injection attacks
- Protecting the data of our web-app services

Timesheet

Week 1 (2/2/24 - 2/9/24)

Shams:
(2/2/24)

- brainstormed coding method instructions (2 hrs)
- formatted & organized notebook structure (30 mins)
- added resume (1 minute)

(2/3/24)

- re-ordered & re-organized structure of the notebook (25 mins)
- created a clean tracking format for the timesheet (3 mins)

(2/4/24)

- Added comments outlining the objective of each plan/page for clarity (4 mins)

(2/5/24)

- wrote out a menu-item-popularity coding method and test-checked, figured a way to build a sample data table with python using pandas, then integrated dictionaries and logic into the method (4 hours)

(2/6/24)

- Pushed file containing methods onto github (5 mins)

(2/8/24)

- Brainstormed Java GUI and database options for the project (10 mins)

(2/11/24)

- Worked on the technical plan document, outlining details of the design and UI of the project (10 minutes)

(2/13/24)

- Worked on the technical summary, adding information for connecting Java to MySQL using JDBC (15 minutes)

(2/14/24)

- Setup our coding environment with different dependencies like spring boot devtools, spring web, maven, and Vlaadin with the VS Code IDE, and created a manager entity and class to store variables and important data relating to the managers' table. Opened pull req for main branch. (4 hours)

Ryan:
(2/3/24)

- Brainstorming & adding points of focus on Software Documentation. (1 hour)
- Implemented Table of Contents & Page #'s. (5 mins)
- Created Github Repo & added readMe. (30 mins)
- Added Resume. (1 min)
- Brainstormed software functionality in scope, pending team review. (1 hour)

(2/4/24)

- Built the basic functionality of classes and simulated data using in-memory data structures, pushed to repository. (30 mins)
- Reverted changes, more planning to do (1 min)

(2/9/24)

- Created & added Gantt Chart to Project Plan document & Github README.md

(2/11/24)

- Drafted Data Management Plan (1 hour)
- Created High-Fidelity UI for Sprint 1 (15 mins)

Nhut:

(2/4/24)

- Added resume (1 min)
- Added guidelines into work-load schedule (10 min)
- Reviewed work-planned schedule from team members (5 min)

(2/14/24)

- Worked on technical summary plan (30m)

Isa:

(2/4/24)

- Added Resume (1 min)

(2/11/24)

- Worked on the Technical Plan (5 min)

Prince:

(2/6/24)

- Added resume (1 min)

(2/11/24)

- Add PDF resume to repo - open pull request (5 min)
- Add gitignore file to repot - open pull request (10 min)

(2/13/24)

- Create a proof of concept demoing a possible architecture of the app. Implementing Spring Boot + MySQL (5 hours)

    - Interacting with the backend via a REST API. All frontend UI is handled client side

Week 2 (2/16/24 - 2/23/24)

Nhut:
(2/16/24)
- Continued to work on technical summary plan (10m)

(2/17/24)
- Worked on test plan summary and technical summary plan (20m)

(2/18/24)
- Added more information to the technical summary plan (5m)
- Worked on Schedule sheet (work breakdown structure and workload plan) (50m)
- Joined virtual group meeting to discuss about plans for final product (10m)

Shams:
(2/17/24)
- Revised Technical Summary and Test Plan with Ryan (3 hours)
- Broke down how Spring Boot will be viable to our software development process with Ryan (3 hours)

(2/18/24)
- Helped work on the work breakdown structure and researched videos on different GUI's we could use for our desktop executable project (1 hour)
- Discussed on call whether we'd make a web app or a desktop exe file. (10 minutes)

Ryan:
(2/17/24)
- Revised Technical Summary and Test Plan with Shams (3 hours)
- Broke down how Spring Boot will be viable to our software development process with Shams (3 hours)

(2/18/24)
- Reviewed Planning Document for Submissions w/ Shams, Nhut & Isa (5 mins)
- Discussed future plans for project on MS Teams (10 mins)

Isa

(2/18/24)

- Added onto the Work Breakdown Structure. (1 hour)
- Discussed on MS Teams with team members about the future of the project and what direction to take. (10 mins)

Week 3: