Kennesaw State University

SWE 3313 Group 3 Project

Professor David Stable

Automation of J’s

This project develops a computerized system to help restaurant personnel coordinate their activities and improve their services, and for the management to track business growth and create future plans.

Ethan Butler

Tuyen Le

Ryan Mobley

Jason Tang

Justin Wade

# Project Plan – Table of Contents

[Scope 1](#_Toc51945475)

[Schedule 3](#_Toc51945476)

[Team Organization and Roles 5](#_Toc51945477)

[Resumes 6](#_Toc51945478)

[Technical Description 11](#_Toc51945484)

[Data Management Plan 13](#_Toc51945485)

[Test Plan 15](#_Toc51945486)

# Scope

1. App: Table Management
   * Overhead view of the floorplan
   * Mark tables as ready, occupied, unavailable, or dirty
     1. Color-coded table images signify status
   * View and place orders by table
     1. Menu items grouped by category, all with easily adjustable options for preparation (add or remove items, change amount cooked)
   * Take payment by table
2. App: Order Management/Queue
   * Order queue will be the primary screen for the kitchen staff
   * Kitchen staff can click on an order and change its status, which updates the floorplan view showing that the table’s order is ready to be delivered. When it is dropped off, the wait staff can mark it as delivered
   * Managers can track the order queue and void/refund items as needed for orders that have been completed
3. App: Task Lists
   * Display checklists and completion status
   * Employees mark tasks completed, and this completed task is recorded
4. Backend: Interface and Insights
   * Database stores and sends information about orders and other relevant info
   * Interface: a Windows application to view insights and manage the system
   * Insights Offered
     1. Hourly and daily statistics
     2. Revenue and revenue percentage per menu item
     3. Menu item popularity
     4. Personnel efficiency (how quickly orders are made and delivered, time between patrons leaving and table available, etc.)
     5. Average customer turnaround time
     6. Average order preparation time
     7. Customer analysis (patron traffic in a selected timeframe)
     8. Finance management (revenue in a selected timeframe)
   * System Management Interface
     1. Visual editor for the floorplan
     2. Menu item management (item name, price, options)
     3. Employee management (add/remove, change roles)
     4. Employee checklists (repeatable, modifiable)
5. User Authentication/Roles
   * Login via employee number
   * Depending on the user’s role, display a different default view, and allow for them to change views
     1. Role (Default view, other optional views)
        1. Amount and location of hardware
     2. Wait staff (Floorplan and order view, task view)
        1. Will each have an individual tablet
     3. Manager (Floorplan, can switch between all views)
        1. Will each have an individual tablet
        2. Computer in an office
     4. Host (Floorplan, order view, waitlist, reservations, task view)
        1. Tablet at the front of the house
     5. Bus staff (Floorplan view, task view)
        1. Tablet at the back of the house
     6. Kitchen staff (Order queue view, task view)
        1. Tablet in the kitchen, additional screens as needed

# Schedule

### Work Breakdown Structure

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task ID | Task | Planned Start | Planned Finish | Planned Workload  (hours) | Actual Workload (hours) | Progress (%) |
| **1** | **Planning** | **9/19/20** | **11/3/20** | **32** |  | **33%** |
| 1.1 | Project Plan Docs | 9/19/20 | 9/25/20 | 16 | 20 | 100% |
| 1.2 | Sprint One Planning | 10/5/20 | 10/6/20 | 8 |  |  |
| 1.3 | Sprint Two Planning | 11/2/20 | 11/3/20 | 8 |  |  |
| **2** | **Documentation** | **11/2/20** | **11/20/20** | **30** |  |  |
| 2.1 | Requirements Docs | 11/2/20 | 11/6/20 | 15 |  |  |
| 2.2 | System Design Docs | 11/16/20 | 11/20/20 | 15 |  |  |
| **3** | **System Design** | **10/5/20** | **11/27/20** | **52** |  |  |
| 3.1 | Data Management | 10/5/20 | 10/16/20 | 16 |  |  |
| 3.1.1 | Database Diagram | 10/5/20 | 10/9/20 | 8 |  |  |
| 3.1.2 | Database Building | 10/12/20 | 10/16/20 | 8 |  |  |
| 3.2 | System Programming | 10/19/20 | 10/30/20 | 20 |  |  |
| 3.2.1 | Connection to App | 10/19/20 | 10/30/20 | 8 |  |  |
| 3.2.2 | Console Interface | 10/19/20 | 10/30/20 | 12 |  |  |
| 3.3 | Improvements | 11/2/20 | 11/27/20 | 16 |  |  |
| **4** | **App Development** | **10/5/20** | **11/27/20** | **54** |  |  |
| 4.1 | UI Design | 10/5/20 | 10/16/20 | 12 |  |  |
| 4.1.1 | Login Screen | 10/5/20 | 10/16/20 | 4 |  |  |
| 4.1.2 | UI Views/Tabs | 10/5/20 | 10/16/20 | 8 |  |  |
| 4.2 | Programming | 10/12/20 | 10/30/20 | 18 |  |  |
| 4.2.1 | Intranet Login | 10/12/20 | 10/30/20 | 6 |  |  |
| 4.2.2 | Views Functionality | 10/12/20 | 10/30/20 | 12 |  |  |
| 4.3 | Improvements | 11/2/20 | 11/27/20 | 24 |  |  |
| **5** | **Testing, Optimizing** | **10/12/20** | **11/27/20** | **36** |  |  |
| 5.1 | Unit Testing | 10/12/20 | 11/27/20 | 12 |  |  |
| 5.2 | Prototype 1 Testing | 10/26/20 | 11/6/20 | 16 |  |  |
| 5.2.1 | QA Testing | 10/26/20 | 10/30/20 | 8 |  |  |
| 5.2.2 | Delivery and UAT | 10/30/20 | 11/6/20 | 8 |  |  |
| 5.3 | Prototype 2 QA | 11/23/20 | 11/27/20 | 8 |  |  |
| **6** | **Production** | **11/30/20** | **12/4/20** | **20** |  |  |
| 6.1 | Finalizing, Delivery to Client | 11/30/20 | 12/4/20 | 20 |  |  |

### Gantt Chart – Project Timeline

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Week Start** | | | | | | | | | | |
|  | 9/21 | 9/28 | 10/5 | 10/12 | 10/19 | 10/26 | 11/2 | 11/9 | 11/16 | 11/23 | 11/30 |
| **Planning** |  |  |  |  |  |  |  |  |  |  |  |
| Project Plan |  |  |  |  |  |  |  |  |  |  |  |
| Sprint One |  |  |  |  |  |  |  |  |  |  |  |
| Sprint Two |  |  |  |  |  |  |  |  |  |  |  |
| **Documentation** |  |  |  |  |  |  |  |  |  |  |  |
| Reqs Docs |  |  |  |  |  |  |  |  |  |  |  |
| Sys Design Docs |  |  |  |  |  |  |  |  |  |  |  |
| **System Design** |  |  |  |  |  |  |  |  |  |  |  |
| DB Diagram |  |  |  | ­­­­↴ |  |  |  |  |  |  |  |
| DB Building |  |  |  |  | ↴ |  |  |  |  |  |  |
| Connectivity |  |  |  |  |  |  |  |  |  |  |  |
| Interface |  |  |  |  |  |  |  |  |  |  |  |
| Improvements |  |  |  |  |  |  |  |  |  |  |  |
| **App Dev** |  |  |  |  |  |  |  |  |  |  |  |
| Login UI |  |  |  |  |  |  |  |  |  |  |  |
| Views UI |  |  |  |  | ↴ |  |  |  |  |  |  |
| Intranet Login |  |  |  |  |  |  |  |  |  |  |  |
| Views Func. |  |  |  |  |  |  |  |  |  |  |  |
| Improvements |  |  |  |  |  |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |  |  |  |  |  |
| Unit Testing |  |  |  |  |  |  |  |  |  |  |  |
| Proto 1 QA |  |  |  |  |  |  |  |  |  |  |  |
| Proto 1 UAT |  |  |  |  |  |  |  |  |  |  |  |
| Proto 2 QA |  |  |  |  |  |  |  |  |  |  |  |
| **Production** |  |  |  |  |  |  |  |  |  |  |  |
| Delivery |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Deliverable Submissions** | |
| Project Plan Documents | 9/25/20 |
| Prototype 1 | 10/30/20 |
| Requirements Documents | 11/6/20 |
| System Design Documents | 11/20/20 |
| Prototype 2 | 12/4/20 |
| Project Presentation | 12/4/20 |

|  |  |
| --- | --- |
| Key |  |
|  | Task |
|  | Dependent Task (↴) |
|  | Completed Throughout |
|  | Task Group |

# Team Organization and Roles

#### Project Manager: Ryan Mobley

#### UI Designer: Ethan Butler

#### Lead Document Writer: Tuyen Le

#### Database and Systems Administrator: Justin Wade

#### Lead Programmer: Jason Tang

# Resumes

# 

# 

  
Jason Tang

1234 Not Real Address | Kennesaw, Georgia 30144

123-456-7890 | jtang14@students.kennesaw.edu

**Objective:**

* Secure a job fit for my skills, increase knowledge of technology and gain experience.

**Education:**

* North Gwinnett High School – (2013 to 2017)
* Weighted GPA of 3.0
* Georgia Gwinnett College (Fall 2017 – Spring 2019)
* Kennesaw State University. (Fall 2019 – Present)
* Current GPA of 3.63.
* **Bachelor of Science in Computer Science**

**Awards & Honors:**

* KSU’s College of Computing & Software Engineering Dean’s List (as of Fall 2019-Spring 2020)

**Skills:**

* Experienced in HTML, JavaScript, Python, Java and C# programming languages.
* Experienced with Office 365 Suite administration.
* Experienced with OneDrive & SharePoint administration.
* Experienced with mobile devices (iOS & Android).

|  |  |
| --- | --- |
| **Justin**  **Wade** | horizontal line **Justin Wade** 1234 Made Up Name Drive  Tucker, GA, 30084  123-456-7890  jwade53@students.kennesaw.edu |
| **ㅡ** **Skills** | horizontal line   * Proficient knowledge of C# * Moderate knowledge of python * Moderate knowledge of C++ |
| **ㅡ** **Experience** | horizontal line **Pollo Tropical /** CashierJuly 2018 - December 2018, Tucker, GA Worked part-time as a restaurant cashier on weekends. Left when the restaurant ceased operations. **Alltech /** QA AssistantJanuary 2019 - July 2019, Tucker, GA Worked part-time assisting the Quality Assurance Manager at a CNC manufacturing firm after school. Left to attend college. **Honey Baked Ham /** CashierNovember 2019 - December 2019, Tucker, GA Worked seasonally as a restaurant cashier for the Thanksgiving and Christmas holidays, **Camp Westminster /** Program CounselorMay 2020 - August 2020, Conyers, GA Worked as a camp counselor full time for the summer.  horizontal line **Kennesaw State University /** Bachelor of Computer ScienceAugust 2019 - Now, Marietta, GA Degree still in progress. **Tucker High School /** High School DiplomaAugust 2015 - May 2019, Tucker, GA Achieved a 4.095 GPA. |

# Technical Description

Our software facilitates and coordinates all major restaurant activities. It acts as a point-of-sale system, table and order manager, and digital checklist. Employees will use Apple iPads connected to the restaurant’s network to serve customers and view table and order status in real time. The software system will be tracking the amount of the average patron traffic based on time of day, date, and day of the week in order to offer insights to management. The backend application will be hosted on a Windows computer in the restaurant, and the iPads will interact with this application to send and receive data. This application will have an interface in which management can view insights and fully modify information within the app, such as the restaurant’s floor plan; menu items, details, and prices; employees and their roles; and checklists items. A database will store much of this relevant information to allow for easy querying of insights like revenue, menu item popularity, and turnaround and preparation times.

Upon opening the app, employees will be met with a login screen where they will input their unique ID. The app will display relevant user interfaces based on their role in the restaurant. If a host/hostess logs in onto the device, the user interface will show the floorplan of the restaurant and what tables are reserved, taken, empty, or in the process of cleaning. Wait staff will order status by table and will receive notifications when orders are ready from the kitchen and when tables need to be checked on. Selecting a table will show more information, including the table’s bill and an option to take a payment. For clarity, wait staff and host staff will see a color-coded status for each table. For kitchen staff, it would show orders in the queue and allow them to mark orders as completed. Bus staff will be notified when tables are dirty, and they will be able to view all tables that require attention.

Within the iPad app, shift managers will be able to oversee all of the tasks at hand and be able to observe any section of the restaurant so that they can ensure that everything is running smoothly. The manager will have an additional view to see which employees are scheduled in a shift. Additionally, managers will have the option to void or refund orders as needed.

This project will require various resources for development, and we expect some restrictions in the development process. We will be developing two connected but separate applications; the app will be developed for iOS and the backend application will run on Windows. This means we will need both Windows and Mac environments for development and testing. We expect some difficulties as we integrate with existing payment processing systems and potentially an online ordering system. We expect that the client’s current restaurant computer running Windows will be able to host the service, and they plan to purchase an appropriate number of iPads to be used by employees during a shift.

# Data Management Plan

In our software, we will need to retain large amounts of data from a multitude of areas. Given that the software requirements include both daily and hourly summaries of revenue both overall and by menu item, popularity of each menu item, efficiency of employees, average turnaround times of customers, and food preparation times, we would need to archive almost all of the data that is inputted into our software. Other insights will include employee attendance and anticipation of customer traffic based on previous data. A properly structured database will be necessary to ensure that information can be easily queried to ensure accurate and fast insights.

The core of our database will be one table that stores all orders. This includes what time the customer(s) arrived, what time they were seated, what time their order was taken, what time they were served, what time they left, and which employee was their server. It also includes how many people were in the party and how many of what items they ordered. This table would be updated with each order when the customers pay their bills. This would be our largest data set where many of our other tables would draw data from in order to output the insights talked about in the first paragraph.

We will have another table specifically for menu items. The information in this table will only be modified by manager actions, and we plan to have it automatically update once a day when the restaurant closes. It will include the name of every single menu item, the cost of a customer to purchase it, the cost to produce it, and the number ordered each hour (based on the orders table). This table would be updated in real-time with the items that were ordered and the quantities, and it would be updated manually when new menu items are introduced, or prices are changed.

A staff table will contain current data for employees. This will include the name of each employee, their employee number, their clock-in and clock-out times, and their hourly pay. It will be updated automatically in real-time to include clock-in and clock-out times and updated through the application when employees need to be added or removed.

Finally, we will have a database devoted to tasks. It will include the name of each task, the time it was started, the time it was completed, and the employee who performed it. It will be updated in real-time. Additionally, we will have certain information not in a database but located in configuration files on the host computer. This will include things like the floor layout and scheduled tasks.

The data in these tables can be queried and modified using the interface in our Windows application. This application will provide a way to view all of the data in easily manageable formats. The user will be able to sort the data in different ways, such as turnaround time overall or by the employee. They will also be able to view certain calculated insights, such as overall revenue or averages of certain pieces of data.

# Test Plan

The system and application will be developed with the intention of unit testing wherever possible to ensure reliable and functional code. Unit tests will be written as functions of the programs are written, and integration tests will be implemented as functional elements of the software are completed and ready to be built. Both unit tests and integration tests will be performed as an automated build step as the software is compiled. The iOS app and Windows backend application will undergo quality assurance testing of their user interfaces and functionalities before prototypes are delivered for user testing.

Our team will dedicate a server for hosting the most recent major build of the backend software for testing throughout the development process. Our testing iPads will interact with this server to send and retrieve information from the database. This way, we can catch any database issues as they arise throughout testing and modification of the software.

Two prototypes will be delivered to the client for user acceptance testing. The first prototype, scheduled for delivery Oct 30, will provide all of the basic functionalities of the software. Our team will assist with the installation and setup of the backend server software and iOS app, and we will confirm that the app is properly sharing data with the system. From the client, we expect feedback regarding usability concerns, defects, and additional feature requirements for the iOS app and the backend application. This feedback will be incorporated as the second sprint is planned.

The second prototype, scheduled for delivery Dec 4, is expected to be fully operational and ready for use after setup. Throughout this second sprint, quality assurance testing will incorporate physical card readers. We plan to have the software built and ready for deployment the week prior so that the week of Nov 30 to Dec 4 can be dedicated to addressing any final concerns and assisting the client with the final setup.