Working with your team

Update the 10 Agile/user stories from Lab 4 to include a detailed description, assumptions, assignee, story point estimation, priority, list of tasks and tests, and definition of done for Sprint 1. There should be only one submission per team from the product owner.

The stories from Sprint 1

1. As an administrator I want to be able to view all restaurants

Assumption: the initial restaurants and menus are in the database and are valid at the start of the program. #11, #15 is completed

a. Priority: 1

b. Estimation: 5

c. Assignees: Dustin/Nick

d. Detailed Description: Administrators can see a single list of all available restaurants and their distances to Saddleback College in the GUI.

e. List of tasks and tests: Extract the restaurant names from the database, create methods in the database manager class to retrieve the distance to saddleback and the total revenue. Test that all restaurants are printed with their corresponding distances.

f. Definition of done: All restaurants, their distances to saddleback and their total revenue are printed to the GUI for an administrator to see. Standard of coding has been established. Code has been commented for future methods to be used when variables are being extracted from server. Testing has been passed. Has been approved by product owner. Code is submitted to Github and documented.

2. As an administrator I want to be able to view menus for a particular restaurant.

Assumption: Database contains the requested restuarant from input and items on restaurant are valid and viewable. Assuming #1 is completed.

a. Priority: 2

b .Estimation: 5

c. Assignees: Nick/Dustin

d. Detailed Description: An administrator can select a restaurant name and then see that restaurant's menu items(item name, price) in the GUI.

e. Tasks and tests: Create methods to get menu items for a specific restaurant, test with different restaurants and print menu items. Print to the GUI elements to display the information there. Print the menu items to the GUI elements and ensure everything looks good.

f. Definition of done: Administrators can view a restaurant’s menu items(name, price) in the GUI. All methods involved have been documented(for doxygen later on). Tests have been passed. Approved by the product owner, code on github.

3. As an administrator, I want to be able to add a new restaurant by user input or by the additional two restaurants from file. Add the name, menu items and its distance(s) to the database.

Assumption: The two given additional restaurants are hardcoded to be added in. Such a database has been established and SQL knowledge is well versed. GUI elements to view restaurants have been implemented and are working. This is assuming Agile Story #1, #2, #15 is also completed.

a. Priority: 2

b. Estimation: 8

c. Assignees: Kev/Adam

d. Detailed Description: Admins can add the 2 given restaurants to the database from file OR can input the data for a new restaurant themselves and add it to the database, after the input data has been validated.SQL Databases are connected to QT, thus having the database be modified by QT commands.

e. Tasks and Tests: Implement methods in the database to add new restaurants and update the distances to other restaurants. This involves creating a restaurant class for consolidation and use in later parts of the project. Add restaurants with given additional two restaurants, to ensure that the add methods in the database class are working. Print the restaurants names to the GUI(implemented in a previous story) to verify that the new Restaurants have been added. Implement user validation with QT and c++ to validate all user input for a new restaurant. Add a new a restaurant(with valid data) using the GUI elements and reprint the restaurants to the GUI to ensure the addition worked correctly. Test with bad data to test (leave fields empty, input strings where numeric data is requested, input duplicate restaurant name etc)the validation parts of the code to ensure that no invalid/unexpected names or prices are added to the database or printed.

f. Definition of done: Administrators can add the 2 additional restaurants from the given data OR can add new restaurants through user input(that is validated). Code is documented and on Github. Tests have been passed and demonstrated to at least one other team member. Approved by the product owner.

4. As an administrator I want to be able to add and remove menu items for current restaurant and the changes will reflect in the database.

Assumption: The items are present within database and modifiable for an admin. GUI elements to view and modify menu items have been implemented and are working. #11, #3 are completed

a. Priority: 4

b. Estimation: 5

c. Detailed Description: The administrator can individually add or remove any menu item by user input for the selected restaurant they chose in the GUI. The changes are reflected in the GUI and the SQL database.

d. Tasks and Tests: Implement methods in the database class to add and remove menu items. Test each with hardcoded data to ensure the database methods are working. Implement GUI elements that allow an Administrator to either remove or add a menu item for a particular restaurant. Add validation with Qt and c++ for validating input and add utility methods to the database to prevent duplicate items and prevent restaurants from having more than 8 menu items. Test input and output with valid names and prices. Invalid input must NOT be updated to the database. The user must be allowed to cancel transaction or re-enter valid input.

e. Definition of done: Administrator can select a restaurant through the GUI and add or remove menu items(via user input in the GUI as long as the selected restaurant has less than 8 items, input is validated) and see their changes in the GUI. All code used is documented and on Github. Product owner approves. Tests have been run, passed and demonstrated to at least one other team member.

f. Assignees: Adam/Kev

5. As an administrator, I want the initial restaurants in the database when the program is started so that the restaurants and their menus are available for visiting.

Assumption: There are valid restaurants and menu items in the SQL database.

a. Priority: 1

b. Estimation: 1

c. Detailed Description: The initial 10 restaurants and their respective menus are in the database at the start of the program

d. Tasks and Tests: Populate SQL Database from input(table for restaurants and table for menu items). Use SQLite studio to add the initial 10 restaurants to the database. Their menus will be added into the database and QT will handle the menu items by storing such items into vectors. Each restaurant will be indirectly tied to a vector which contains the menu items.

e. Definition of done: The first 10 restaurants and their menus are in the database ready for use. Data is checked over by all team members and approved by product owner. The database file with all initial data(is valid) is on Github.

f. Assignees: Nick/Dustin

6. As an administrator, I want to be able to use a GUI Interface so that I can make changes to the item price on any menu and/or add new restaurants. (GUI)

Assumption: The GUI is easy to use (user-friendly) and functional. #11 is complete.

a. Priority: 3

b. Estimation: 13

c. Detailed Description: The user interface must allow the administrator to make any necessary changes to the price of the items on selected menu. This interface must also have an access point to allow the admin such that they are able to add new items to the menu.

d. Tasks and Tests: Create the GUI in Qt and ensure the tab order is correct, name each widget based on its location in the window and functionality. Add validation code where necessary.

e. Definition of done: GUI is implemented and working as intended, all of its validation has been implemented and tested. The appearance is approved by product owner. Documentation of the variables, along with comments used along the GUI appearance methods are detailed. Each member has been walked through the GUI so they can work with it if need be. Code is on Github

f. Assignees: Dustin/Nick

Sprint 2 stories

1. As a saddleback student I want to be able to travel from restaurant to restaurant
   1. Assumption(s): All data is in the database and valid
   2. Priority: 1
   3. Estimation: 13
   4. Assignees: Dustin/Nick
   5. Detailed description: A saddleback student iterates through the vector of restaurant objects and see their menus
   6. List of Tasks and Tests: Create Restaurant classes. Read all relevant data from the database and fill the classes with data. Start at the first restaurant in the db for now. Run through and validate each restaurant.
   7. Definition of done: A Saddleback student will open the program and see the home page. From there, they can select the default trip button(More trips will be added later) and be taken to the trips tab to move from restaurant to restaurant in a predetermined order.  
      Note: The restaurants will be visited in the order they are stored in the database, the functionality to visit in the most efficient order will be added at a later time, for now we start at the first restaurant. The trips tab will be hidden once all restaurants have been visited and the review tab will appear. The user can then hit the review to home button to return to the home page. Trips will be persistent in the future.
2. As an administrator, I want the administrator functionality to only be accessible by other administrators.
   1. Assumption(s): The admin tab is only available to admins at start of program.
   2. Priority: 1
   3. Estimation: 2
   4. Assignees: Adam/Keval
   5. Detailed description: An administrator can click the login button and enter the password, if they get the password correct, the admin tab will appear
   6. List of Tasks and Tests: Create ui elements to gather the password from an administrator. Create code segments to verify the password and notify the administrator if they entered the correct password. Show the admin tab if the password is correct
   7. Definition of done: The admin panel is hidden when the program launches. An administrator can click the login button and the password. If they get the password correct, the admin panel will appear. They will be notified if they don’t get the password correct. They can logout as well and hide the admin panel.
3. As a saddleback student, I want to see how much I spent at each restaurants and a grand total spent after I finish a trip
   1. Assumptions: The vector of restaurants is filled, #1 is done
   2. Priority: 2
   3. Estimation: 2
   4. Assignees: Keval/Adam
   5. Detailed Description: At the end of a trip, the trip tab will close and the review tab will open. The review tab will have a table that shows each restaurant and how much was spent at that restaurant. It will also display a grand total spent throughout tab. Purchases will also be reflected in the database.
   6. Lists of Tasks and tests: Accumulate the totals as items are bought, create a table widget and fill it with the restaurant names and their total revenue from the trip. Ensure that the changes are reflected in the database and review table
   7. Definition of done: A saddleback student can look at the review table on the review tab and see how much was spent at each restaurant as well as the grand total for the entire trip once the trip has concluded.
4. As a saddleback student, I want to be able to buy items while I am at a restaurant
   1. Assumptions: #1 is complete
   2. Priority: 1
   3. Estimation: 6
   4. Assignees: Nick/Dustin
   5. Detailed Description: A saddleback student can select a menu item for whatever restaurant they are at, pick a quantity and buy that item. The revenues will be updated in the objects and the database
   6. List of tasks and tests: Create gui elements to let the saddleback student select an item and quantity. Test with various restaurants and ensure the changes are reflected in the database and the trip review tab
   7. Definition of Done: Saddleback student can see their items added to the receipt table when they buy them(they can buy multiple of a particular item if they want) and see the money they spent reflected in the review tab and the grand total indicator on the trip tab as they visit the individual restaurants.