IT314 Software Engineering

Lab-4 Report
Central Mess Management Software
Group: G32

Members:

202001444	PARMAR GAURANG GAJENDRABHAI
202001454	KANANI DEEP MUKESHBHAI
202001075	RAJPUT VRUND ASHWINKUMAR
202001418	RAHIL HARDIK SHUKLA
202001053	KRISH GULABANI
202001443	LIMBACHIYA OM
202001232	PRAJAPATI PARTH KANUBHAI
202001427	MOHAL RAJYAGURU
202001435	MEHTA NEELKUMAR HIRENKUMAR
202001451	VIVEK GODHASARA
202001417	DHANANJAY A VORA

Tools, Technologies and Framework:

(a) You must finalize/identify the tools, technologies, and frameworks you will use to develop/implement your project.

1. Tools : Visual studio Code, Git and Github
2. Front-end development : HTML5, CSS, JavaScript and Bootstrap.
3. Back-end development : JavaScript, Node JS with Express JS (a web framework for Node JS).

(b) For your project, you have to use the NoSQL databases of your choice strictly, and you can also explore and use ElasticSearch (DB) Database for the same.

Database:

MongoDB and mongoose

Reason:

- Popular NoSQL database MongoDB is developed to be adaptable, scalable, and powerful. Applications that are required to manage big volumes of data and need quick read and write operations may consider using it. Its data structure, which lacks a database schema, enables modifications over time without the need to change the database schema. In addition, MongoDB offers a robust query language, a variety of data formats, and indexing possibilities.

- A Node.js and MongoDB object data modelling (ODM) library is called Mongoose. It facilitates the management of data relationships, offers schema validation, and translates between the representation of objects in code and their MongoDB counterparts.

(c) Estimate the effort of your project and narrow down the scope based on the estimation. You can apply Function Point (for user stories) or Use Case Size Point (for use cases) for estimation.

Function Point (for user stories)

User Input:

- Username
- Password
- Ratings
- Overall experience feedback
- Additional comments or suggestions

- Updated quantity of ingredients
- Updated expiry dates of ingredients
- Date and time of starting and ending of subscription
- Cadet information (name, age, gender, etc.) to be added
- Cadet information (name, age, gender, etc.) to be removed
- Admin inputs customer information (name, contact information, etc.) into the system.
- Admin selects a customer to be removed from the database
- Changes to be made to menu items
- Payment details
- Admin selects the date range and time period to filter the statistics data.

User Output:

- (Un)Successful login message
- Previous billing history
- Account information
- Confirmation message after submitting feedback
- List of raw food items in stock
- Quantity of each item
- Expiry dates of each item
- Total revenue
- Display of the current menu with all available food items
- Error message when required information is missing or invalid.
- The filtered data based on the selected date range and time period is displayed to the admin.

User Inquiries:

- Can I view my account details?
- Will my feedback be anonymous?
- How will the management use the feedback provided?
- How often is the inventory data updated?
- Can I view the history of payments?
- How to add new food items to the menu?
- How to remove food items from the menu?
- Can I edit customer information after it has been added to the database?
- Can I search for customers by name or contact information?
- How do I know if a customer has already been added to the database?
- How often is the statistics data updated in real-time?

Internal Logic Files:

- User database with user ID and password
- Billing history database
- Feedback database
- Inventory database
- Add and update functions for inventory
- Payment database with fields for payment
- Cadet database to store cadet information
- Deletion function to remove cadet information from the database
- Menu database, containing all food items

Calculation of the FPs:

Measurement parameters	Low	Avg	High
External Inputs (EI)	3	4	6
External Output (EO)	4	5	7
External Inquiries (EQ)	3	4	6
Internal Logical Files (ILF)	7	10	15
External Interface Files (EIF)	5	7	10

Measurement parameters	Low	UFP for Low category	Avg	UFP for avg category	High	UFP for high category
External inputs (EI)	7	21	6	24	2	12
External outputs (EO)	4	16	6	30	1	7
External inquiries (EQ)	5	15	4	16	2	12
Internal files (ILF)	4	28	5	50	0	0
External interfaces (EIF)	0	0	0	0	0	0
Totals for category		80		120		31

Count-total = 231

Calculation of ∑(fi) :
Here is the rating of all the 14 questionnaires on the scale of 0 to 5.

Scale varies as follow

0 - No Influence
1 - Incidental
2 - Moderate
3 - Average
4 - Significant
5 - Essential

Sr. No.	Questions	Rating
1	Does the system require reliable backup and recovery?	5
2	Are data communications required?	1
3	Are there distributed processing functions?	2
4	Is performance critical?	5
5	Will the system run in an existing, heavily utilized operational environment?	2
6	Does the system require online data entry?	5
7	Does the online data entry require the input transaction to be built over multiple screens or operations?	4
8	Are the master files updated online?	4
9	Are the inputs, outputs, files or inquiries complex?	4
10	Is the internal processing complex?	3
11	Is the code to be designed reusable?	5
12	Are conversion and installation included in the design?	3
13	Is the system designed for multiple installations in different organizations?	0
14	Is the application designed to facilitate change and ease of use by the user?	4
	∑(fi)	**42**

CAF = 0.65 + (0.01 *∑(fi))
= 0.65 + (0.01*42)
= 1.07

FP = Count-total * CAF
= 231 * 1.07
= 247.17