Group Assignment for Introduction to Software Engineering

FOOD CITY SYSTEM

(Final Project Report – Phase I)

Group No.:

Computer Science(UCD) Degree Program

NSBM Green University of Sri Lanka

January 2018

Module: Introduction to Software Engineering

Module Code: Coursework-BMIS202-SE

Project Title: Food City System (Final Project Report – Phase I)

Group Members:

No.	Name	Uni. ID	E-mail	Signature
1	ManoRanjana De Silva	BSC-PLY-COM-16.1-026	manoranjana@smartvalley.lk	
2	Sankalpa Sudheera	BSC-UCD-CSC-16.2-024	sankalpa@smartvalley.lk	
3	Achintha Dileesha	BSC- PLY-COM-16.1-120	achintha.dilesha@gmail.com	
4	Pubudu Dananjaya	BSC- PLY-COM-16.2-202	pubududananjaya@gmail.com	
5	Fernando Y W Asiri K	BSC- PLY-COM-16.2-079		
6	Duleesha Premarathna	BSC-PLY-COM-16.2-222		
7	Kalana Fernando	BSC-PLY-COM-16.2-079		
8	Rajitha Bandara	BSC-PLY-COM-16.2-055		
9	Schintha K	BSC-PLY-COM-16.2-145		
10	Buddhi Perera	BSC-PLY-COM-16.2-199		

Name of the Module Leader: Mrs. Kalpana Weerasinghe

Signature:

Date:

Acknowledgement

We had to take many guidance & advices from some experienced people in writing this documentation. Specially We would like to show our respectable thanks to Mrs. Kalpana Weerasinghe, Lecture of Introduction to Software Engineering at NSBM Green University.

She not only gave us good guidelines but also, she came with many grate attitudes to courage us always in this project to complete it on time. We are also extremely grateful because we managed to complete our assignment before the deadline announced by Mrs. Kalpana.

As well as we would like to share our great gratitude to Mr. Edirisinghe, the owner of the business of our scenario for providing us an unbelievable support to make this project successful.
Really this assignment would not be accomplished without the great effort & co-operation from our group members.

Finally, we would like to express our gratitude to our parents, batch mates & all non-mentioned respondent parties who gave their support & willingness to spend sometimes with us directly or indirectly to complete our project successfully more than we expected.

Table of Contents

1. Introduction

This report is given for the 1st phase of the system for the proposed software of “Food City System”. The report will include almost all the requirements of the system, design patterns and the sample user guide with explanations on requirement.

1.1 Project Overview

This project is subjected to a small-scale business of Mr. Sanjaya Edirisighe. Mr. Edirisinghe is an owner of Food City network in Sri Lanka. He has currently 4 Food Cities over the country and he is willing to spread his business in large scale in the future.

As a noble customer, Mr. Edirisinghe was met and he asked for an easy software solution from our company to develop. He told us his requirements and we could easily adapt to them according to requirements.

In accordance with his requirements, we decided to run this development on Agile methodology. Because the client requested to adapt new functionalities to the software in the future.

1.2 Background

This project is totally implemented using Agile methods and all methodologies are based on it. This is a Java SE based development. Those decisions were taken considering the client’s investment level, knowledge level and other main factors.

The development was done with Java Standard Edition and Data bases were developed based on My SQL.

1.3 Scope of the System

Project covers all the requirements of the client with all strategies of the system. According to the scenario, software is based on 2 major characters as follows:

1. **Administrator**
2. **Employee**

- ✓ This project is a web and mobile based development & developed using Agile Development Model as a Rapid Development.
- ✓ Administrator has the maximum privileges of the system.
- ✓ Administrator can handle all the employee's as well as customer details and he also responsible to add cashier level users to the system.
- ✓ Employee has limited access to the system, but he is responsible to manipulate most risky things within the system like adding a loyal customer, adding suppliers to the system, handling easy payments, generating good receive notes, returning invoices.
- ✓ In this level, there are no special security functions to the system, but users can change their passwords if needed.
- ✓ Also access level is decided once the user enter their user ID to the system. System can understand the user level throughout the user ID.
- ✓ Admin button is automatically disabled, if the logged in user is an employee.
- ✓ All interfaces have been designed in fixed monitor size.
- ✓ Interfaces are designed based on the user level. Administrator has been privileged to access all the interfaces created by employees.
- ✓ Administrator can retrieve auto generated reports throughout the interfaces.
- ✓ Employee can handle major interfaces, but he is not authorized to do critical actions such as removing users, deleting users.
- ✓ Employee is eligible to handle the inventory throughout the system interfaces.

1.3.1 Frontend Scope

Frontend of the software is driven on following areas:

- Customer
- Supplier
- Items
- Donations
- Complains
- Dispose
- Petty cash
- Invoice
- Good receive notes
- Invoice return
- Good receive notes receive Accounts

1.3.2 Backend Scope

Backend of the software is driven on following areas:

- Employee
- Customer stage
- Employee attendance
- Employee positions
- Advertising pay
- Set ETF/EPF
- Employee salary pay
- Reset employee password
- Employee access level
- Employee salary advance
- Bill type

1.4 Objectives

From this development phase, we hope to overcome with several goals.

- Manipulating loyalty customers
- Manipulating employees
- Inventory handling

- Managing supplier information
- Handling easy payments
- Track of user logs
- Generate reports (Daily, Weekly, Annually)

1.5 Future Development

Currently developed software is based on innovative future plans related to our scenario.

Some of them are as follows:

- Issuing loyalty card to the customer based on consuming level
- Issuing printed bills to the customers
- SMS Alert system for promotions, due date notifications for easy payments
- Online payment method adaption to the system
- Purchase goods online from Smart
- Adding Bio-metric Functionalities
- Other new approaches

1.6 Methodology

Typically, software is developed under different methodologies such as *Waterfall, Spiral, Agile* models.

But this software should be developed as this software can be able to develop considering further requirements. Since we have to update the software on client's requirements in the future, we decided to use Agile model and its methodologies.

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and evaluated continuously so teams have a natural mechanism for responding to change quickly.

Agile' s emergence as a huge global movement extending beyond software is driven by the discovery that the only way for organizations to cope with today's turbulent customer-driven marketplace is to become Agile. Agile enables organizations to master continuous change. It permits firms to flourish in a world that is increasingly volatile, uncertain, complex and ambiguous.

Following diagram is showing how we use Agile methodology for this software project in brief.

Figure 1: Agile Method Cycle

According to the above diagram, it is easily understandable that the development uses life cycle iteration on development.

In this way we can easily adapt to the system development and connect new add-ons to the system cost efficiently.

2. Requirement Specification Analysis

In the Food City System, we have to consider a lot of things prior to launch the 1st phase of the development. Mainly the software development was carried out depending on the feasibility study.

Then the requirements were divided into 4 major sections as follows:

- Functional Requirements
- Non-Functional Requirements
- System Requirements
- Domain Requirements

2.1 Feasibility Study

Feasibility study was a most important point in our development. Because through this analysis we could be able to determine the draw backs of the manual system and what can we do making this into computer-based system. Also, how can we adapt the user into the new system and what are the constraints of the client.

The feasibility study was done covering almost all the requirements of the system as well as client's requirements. Typically, we considered about following feasibility types before engaging with this project.

Study area on Feasibility	**Comments**
Investment	Firstly, we considered about the investment plan of the client can weight for this project. Accordingly, we decided the solution plan we could give.
Technological	We studied about the IT knowledge of handling our software. This analysis was done with the employees face to face discussions.
Resources	Since this is an agile development, this will require some additional requirements such as internet connectivity and other pro resources. We surveyed every shop of the client and the ability to access these requirements.
Operational	We also analyzed the ability to face impacts in developing the project in the development background
Legal	Considered about the legal acknowledgement between us and the client
Schedule	In every step, we considered about the schedule feasibility to measure the times we could deliver the project phase by phase.

2.2 Functional Requirements

In this scenario we have to cover up several functional requirements which are available to run as the system to keep the efficiency and usability of the system in peak level.

Several major functional requirements can be detected when the system is considered. Typically, the technical details, calculations and other operations related to the system side is described as Functional requirements.

2.2.1 Main Functions of the System

1. Login to the System under user level
2. Change password
3. Add data to the system
4. Retrieve data from the system
5. Giving feedbacks
6. Error messages and responding

2.2.2 Expanded Functional Constraints

- ✓ Users can access to the system with login privileges
- ✓ User levels are decided by the system according to the login ID
- ✓ Any user can change their passwords, but user IDs are generated by the system
- ✓ Admin button is only enabled once the user access using admin user ID
- ✓ Admin user can create employee IDs as required
- ✓ Default password is generated by the system for new users
- ✓ Default password should be changed at the 1st attempt of login into the system
- ✓ Admin user can access almost all the data generated by employee(cashier)
- ✓ Employees(cashier) can get limited actions within the system

2.3 Non-Functional Requirements

Many non-functional requirements have to be considered within the Food City system which support as behavioral or performance properties that the system must have.

Among them the Performance, Usability, Reliability, Accessibility, Availability, Flexibility, Security, Testability, Maintainability, Efficiency can be mentioned.

- ✓ **Performance**

  The system we implement is running under smooth loop & very stunning. High performance of the computer is not compulsory to run the system. Only Java Virtual Machine is requested run this standalone software.

- ✓ **Usability**

  The usability of the currently developing system was very important. According to our design, the application's interfaces were designed to be much easier & user friendly. The layouts, short-cuts and overall the interfaces have been designed with common icons, native actions etc.
  Human Computer Interaction(HCI) concepts were highly considered in designing eye catching interface blocks.

- ✓ **Security & Reliability**

  We assure that the 1st phase of this standalone application can meet their maximum efficiency levels. Also, the system will be developed by fixing almost all errors with high security.
  Databases are encrypted with secured encryptions to make sure the protection of data stored in the system database.
  Password recovery option is secured by security questions as standard levels.
  Unauthorized user actions have been banned.

- ✓ **Maintainability**

  This requirement is also ensured by the us. System will be maintained by our team time to time. Minor bugs can also be debugged by the users. Overall maintainability is easy in the system since the application developed using Agile Development Model. Overall development can be considered as a Rapid Development.

- ✓ **Accessibility**

  System can be accessed using proper login constraints. It's easy & save the time. Using Cloud Platforms, data can be accessed anytime. It is facilitated to update & retrieve data when internet connection is available in the future development.

- ✓ **Efficiency**

  This requirement is accomplished throughout the system. To increase the efficiency, it will be used very responsive error free methods implement the system.

2.4 System Requirements

In the requirement section, System Requirements are also an important part. This can be divided into 2 main categories as Software Requirement & Hardware Requirement.

2.4.1 Software Requirements

Requirement	Environment/Framework	IDE
Interface Designing	Java FX	NetBeans
Backend	Java	NetBeans
Databases and Connectivity	My SQL	Virtual Localhost server(Wamp)
Illustrations	Online	Drsaw.io
Photo Editing	Graphical	Adobe Photoshop
Documentation	Text Editing	Microsoft Word, Adobe Acrobat Reader

2.4.2 Hardware Requirements

Requirement	Hardware Environment	Unit
Project Simulation	Java Installed Windows 8 or greater OS, 64bit, 100GB HDD, 2GB RAM(Min)	Windows PC

2.5 Domain Requirements

Domain requirements focus on the domain specific constraints of the scenario. Most of the domain requirements have been mentioned on Functional and Non-Functional Requirements.

Especially, following factors have been considered as domain specific requirements of this software development:

- ✓ Limited number of employees have privileged to access the system
- ✓ Loyalty customers are filtered considering consumption
- ✓ Easy payments are only allowed for Credit Card holders or reliable customers on another basis
- ✓ Customer complaints are sent to the Administrator as bulk (no single)

3. Software Design

Software design is a most important chapter in this 1st phase final report. Because, Software design evaluates the stability of the whole software. This section can be divided into 3 sub topics as *Software Architecture, Architectural Design Specifications, Wireframes and User Interfaces.*

3.1 Software Architecture (MVC Model)

In this software development, we used Model-View-Control (MVC) Architecture and its pattern throughout the design. The main reason we decided to use MVC Architecture to this development is that this is an Agile and Rapid application development.
Since we use MVC Architecture, it's very easy to add new things to the software in the future development.

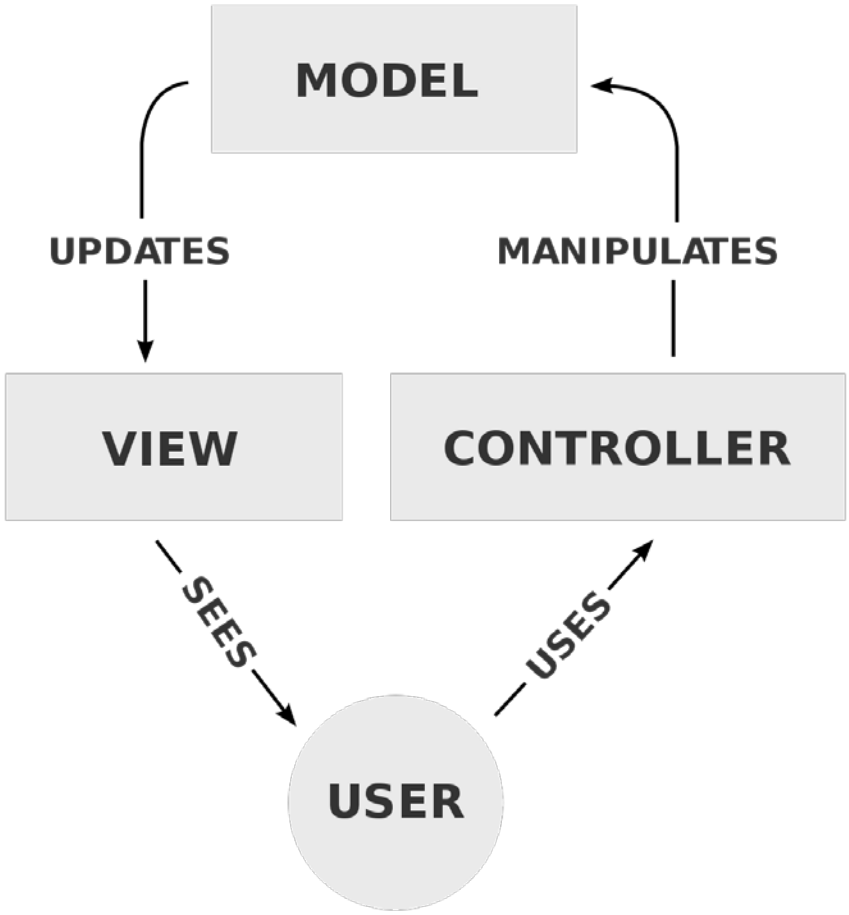


Figure 2: MVC Architecture

According to the Figure:2, it shows the structure of the MVC architecture we used in this software design. In the MVC architecture we used, have 3 major components as *Model*, *View* and *Controller*. They have been used in our development as follows.

Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data.

Objects like customer, employee, items can retrieve, manipulate, update and delete its related data from the database, back to the database or use them to render data.

View

The View component is used for all the UI logic of this software application. For example, the Admin view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, in our system, the *Admin* controller will handle all the interactions and inputs from the *Admin View* and update the database using the *Admin Model*. The same controller will be used to view the *Admin* data.

3.2 Architectural Design Specifications

Since this project development is an Agile, Rapid development we didn't go to complex diagrams. To analysis the system specifications and Work Breakdown Flow, Use Case Diagram was only sketched.

Typically, the designers and the developers of our team adapts with the user requirements and system specifications through Use Case Diagram for Unified Modelling Language(UML).

Also note that our team and the company do not use traditional complex diagrams for the Agile and Rapid application development like this as we follow modern patterns for development. Because, software will be updated and upgraded on different requirements frequently.

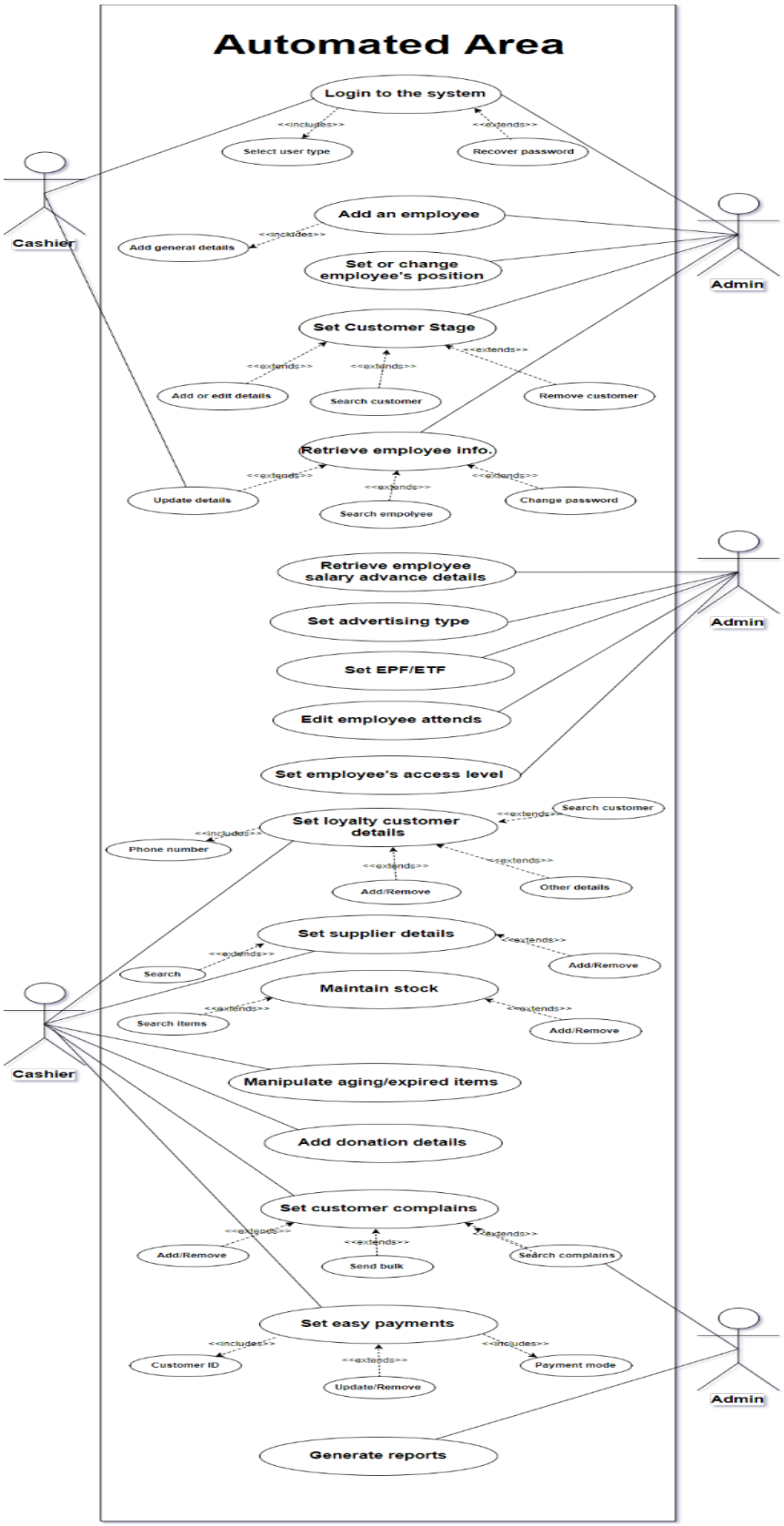


Figure 3: Use Case Diagram

According to the *Figure 3*, almost all the major and minor activates have been illustrated based on software design for the phase 1 development process.

This diagram can be changed on the next phases or on the attempts of additional functionality requests by the client.

3.3 Wireframes

Considering the Use Case Diagram, we designed the blue prints of the software we are about to design. They were drawn as hand sketches to arrange the blocks on better places in each interface. In designing imaginary sketches, we have used Human Computer Interaction(HCI) principles.

Main finalized wireframes we have drawn as follows:

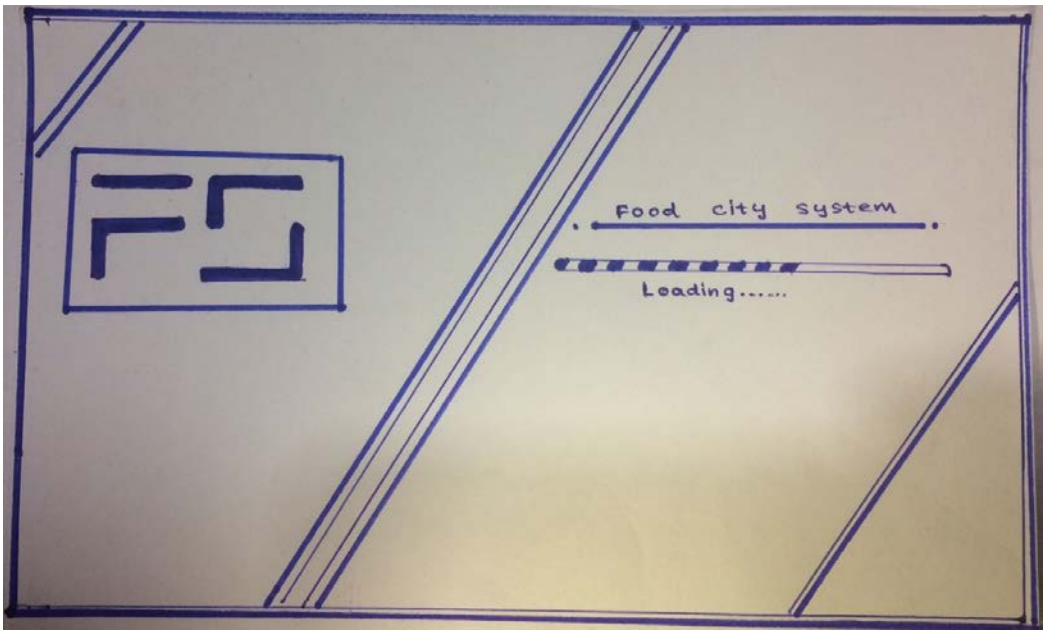


Figure 4: Software Opening Prompt

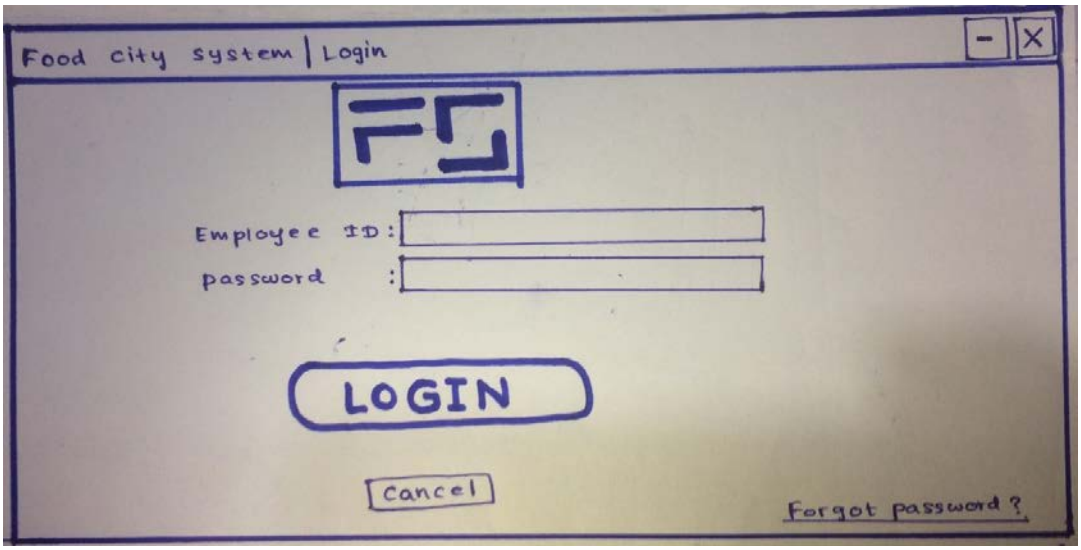


Figure 5: Login Prompt

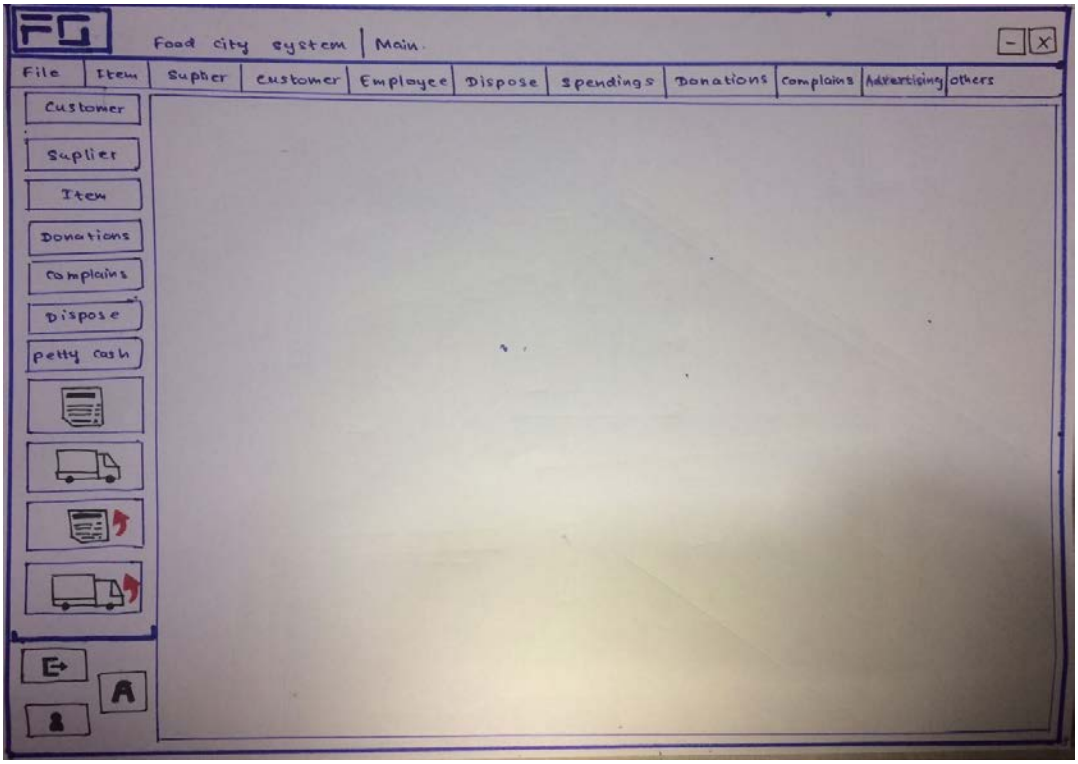


Figure 6: Main Interface

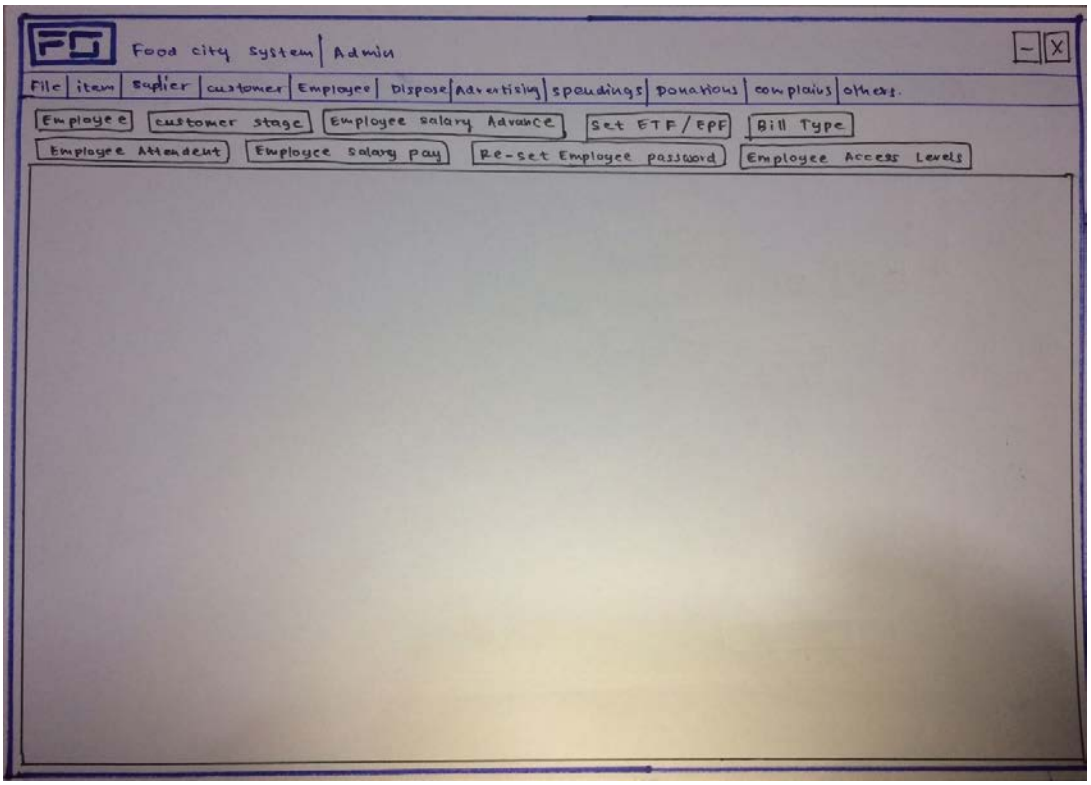


Figure 7: Admin Panel

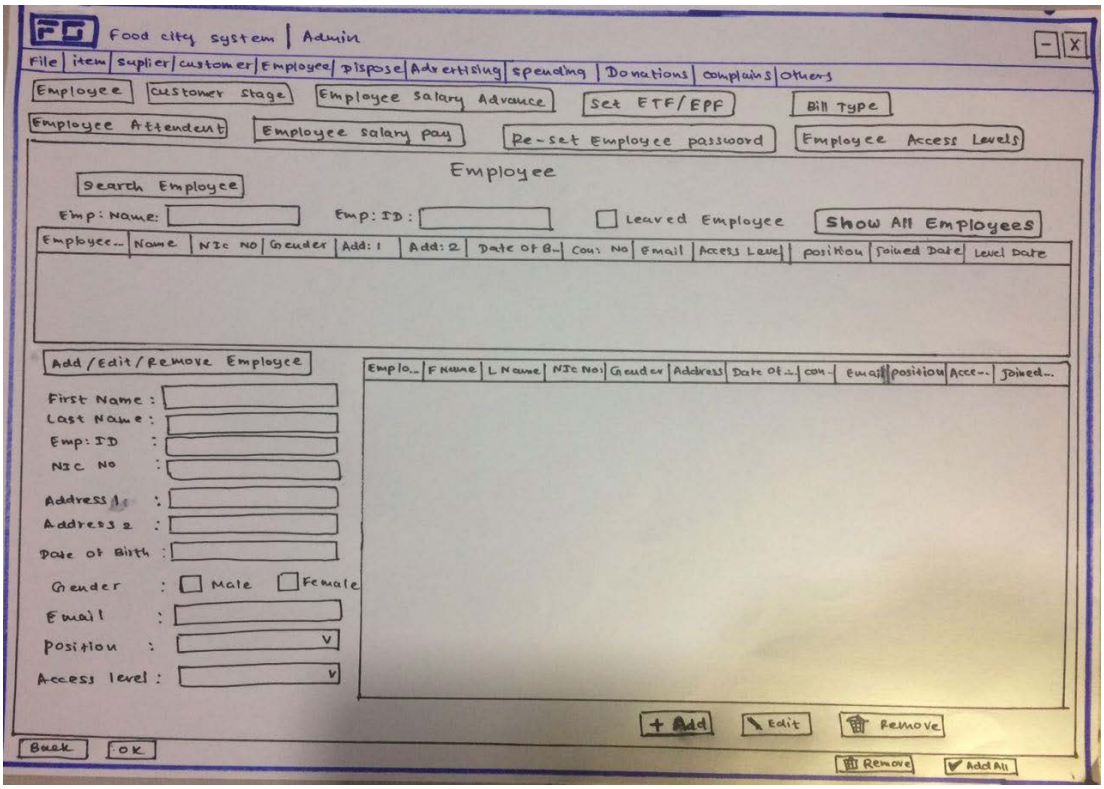


Figure 8: Admin Panel - Adding new employee

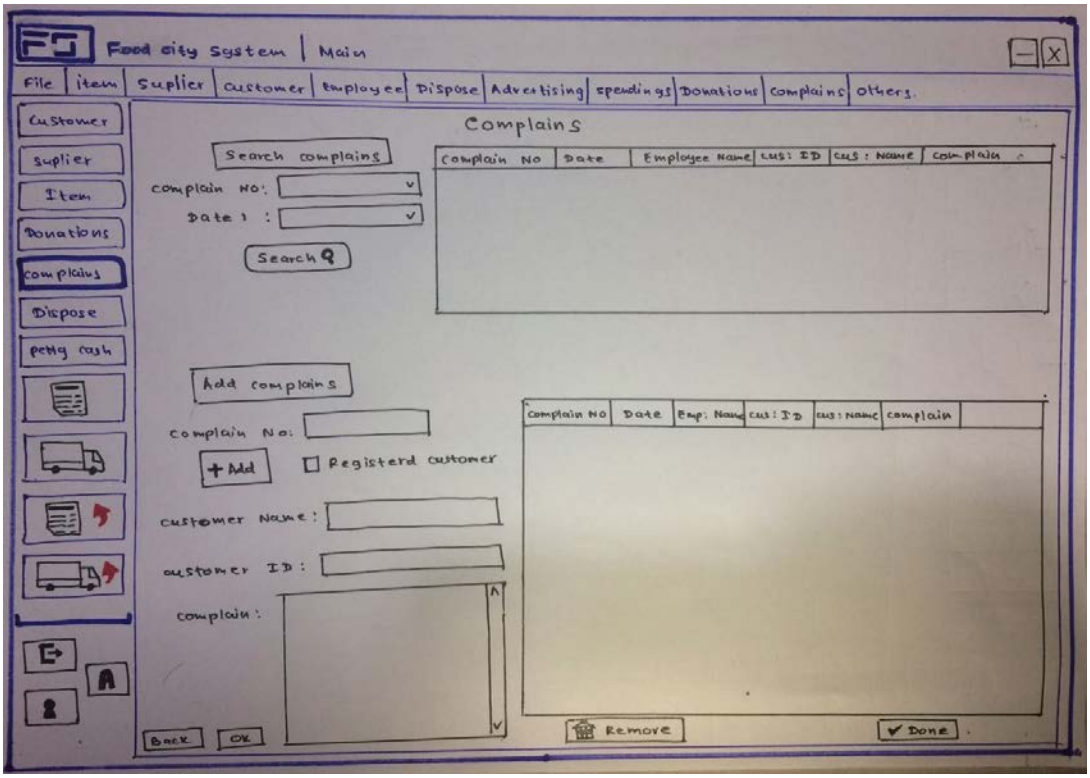


Figure 9: Employee Panel - Adding a customer complain

3.4 User Interfaces

In the development phase 1, all the required interfaces have been designed based on wireframes. They are partially functional since they have not a strong integration between each components and database in this phase.

But most of the common interfaces are functional with feedbacks and error messages on wrong actions taken by the user or auto generated messages by the system.

Few of implemented main user interfaces will be described to ensure the quality of our UI designs for this development.

3.4.1 Login

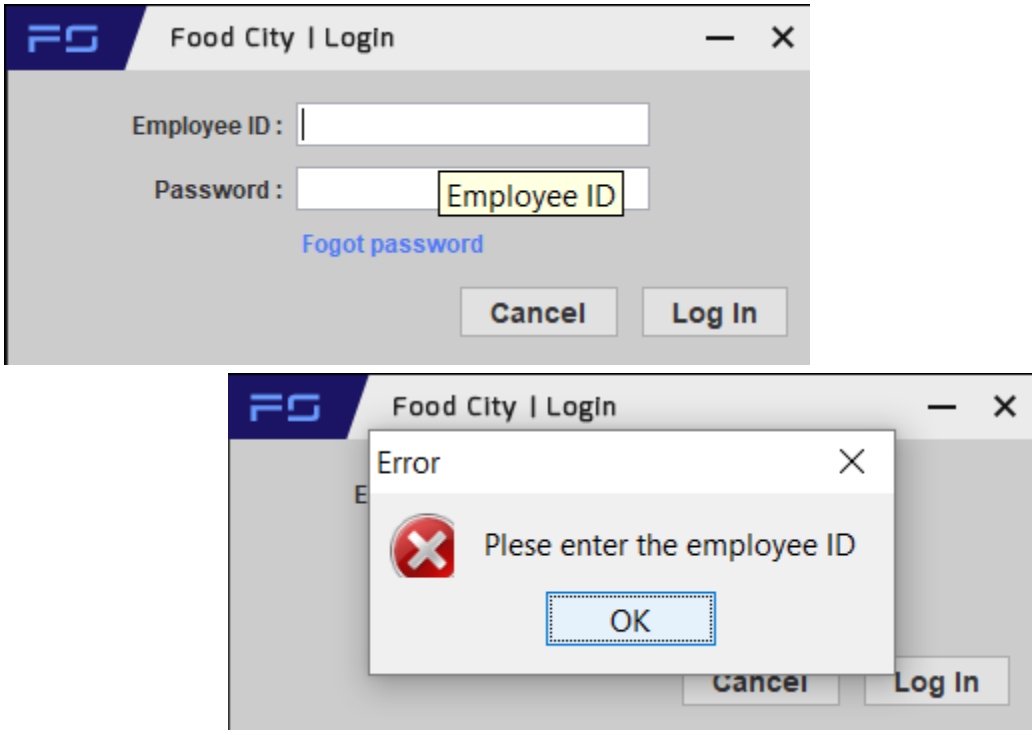


Figure 10: Login Prompt

Figure 4 shows the Login UI screenshot of the Food City system. As we implemented this screen, user(employee) can not be accessed to the system without entering valid user ID. It ensures the security and the reliability of this critical system.

3.4.2 Employee – Admin Interface

Admin Interface is also another major part of our system. It's employee adding section is completed by this phase.

When the admin user tries to add a new employee, he can not skip any required field. Following figure shows how the error message pops up when the user forgets to enter required field.

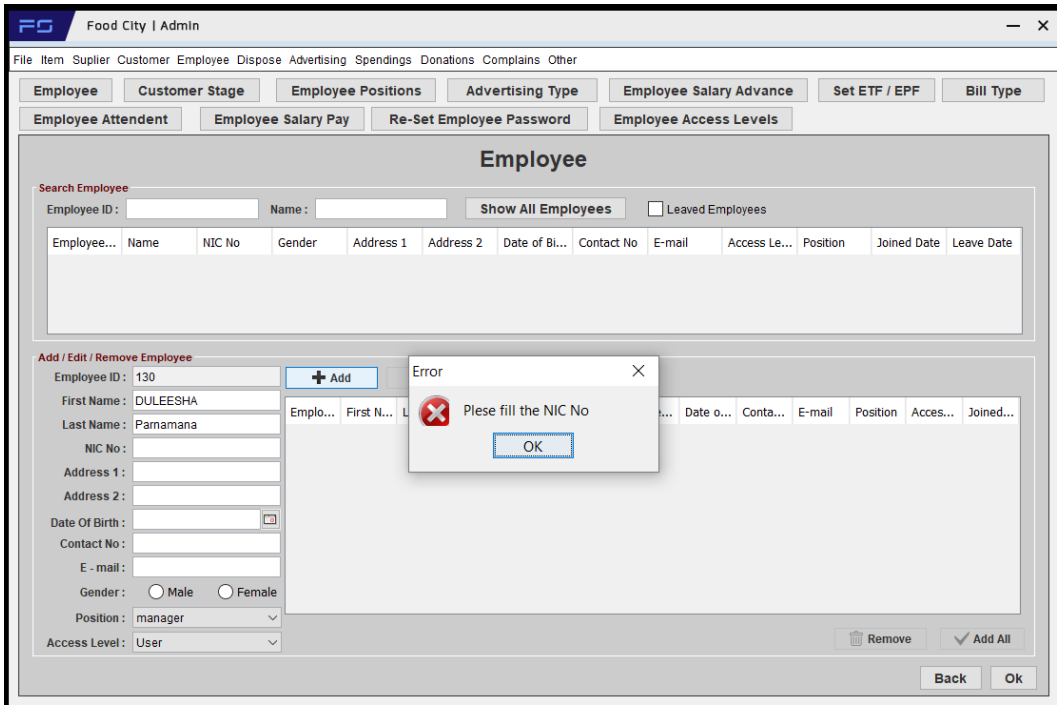


Figure 11: Employee - Admin Panel

3.4.3 Forgot Password

Figure 12: Forgot Password Prompt

According to the above figure, the user cannot skip question challenge against to proceeding in the forgot password process.

Please note that this report only includes the functional interfaces by now for this section. All other interfaces will be described in the *User Guide* section of this report.

4. Testing Evaluation

Testing evaluation is a major part of any phase of software development in our company. The team of us focus on main strategies considering the system according to its operational level and moment.

4.1 Test Plan

For the test plan, we follow *IEEE 829 Standards*. To launch the test plan, we consider all the things we texted on this report. Accordingly, we passed some tests for implemented areas with backends as follows.

#	Task per attempt	User1	User2	User3	User4	User5
1	Login	✓		✓	✓	✓
2	Open Admin Panel	✓	✓	✓	✓	✓
3	Forgot Password	✓		✓	✓	
4	Add New User	✓	✓	✓		✓
5	Add Item	✓		✓		✓
6	Add Complain	✓	✓		✓	

Results

- 4 out of 5 random users performed 1st task correctly
- 5 out of 5 random users performed 2nd task correctly
- 3 out of 5 random users performed 3rd task correctly
- 4 out of 5 random users performed 4th task correctly

- 3 out of 5 random users performed 5th task correctly
- 3 out of 5 random users performed 6th task correctly

Decision

Most users can use the system correctly through desired interfaces. But some generic tasks should be more improved.

4.2 Test Plan Strategies

As our software development is in the phase 1, we used a set of strategies in testing to overcome with the satisfaction of its functionality. Here are some of our major strategies we cared on completing this phase.

4.2.1 Box Testing Approach

Box testing approach is a traditional manual testing method used in software designing. There are 3 main box approaches as *White-Box, Black-Box and Gray-Box.*
White-Box testing is usually used for internal coding and methods deeply known systems. Black-Box Testing is used for internal factors unknown system. Also, the Gray-Box is used for partially known systems.

Since our system is partially know, that derives that some of the parts have been coded we used Gray-Box testing approach for this phase of this development.

Gray-Box Testing

Figure 13: Gray-Box Testing

Gray box testing, also called gray box analysis, is a strategy for software debugging in which the tester has limited knowledge of the internal details of the program.

This approach is a hybrid approach in combining Black-Box and White-Box. Internally known parts can be handled as White-Box testing while internally unknown parts are handled as Black-Box testing.

In this phase, we also have internally completed few of interfaces. Therefore, we have used this approach for testing.

4.2.2 Test Levels

Test levels are also another traditional approach of testing software designs. Typically, the test levels follow the steps as *Unit Testing, Integration Testing, System Testing & Acceptance Testing* in ascending order.

But this development is in the primary age, we couldn't pass all the levels of the test levels.

In this phase, we could only pass *Unit Testing* level.

Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often automated.

In our software, we tested each completed component in an isolation to ensure that they individually work properly.

4.2.3 Interface Testing

Interface Testing is performed to evaluate whether systems or components pass data and control correctly to one another. It is to verify if all the interactions between these modules are working properly and errors are handled properly.

The interface testing was done to the implemented parts of the system at this stage.

In interface testing, we ensured following check lists:

- ✓ Verify that communication between the systems are done correctly
- ✓ Verify if all supported hardware/software has been tested
- ✓ Verify if all linked documents be supported/opened on all platforms

- ✓ Verify the security requirements or encryption while communication happens between systems

4.3 Test Design Specifications

Test Design Specifications are also given following the IEEE 829 Standards and considering the Use Cases of our system. Test Design Specifications will be delivered as a new documentation. In this section we considered following factors of our software development:

- ✓ Test Design Specification Identifiers
- ✓ Features to be tested
- ✓ Approach refinements
- ✓ Test Identification
- ✓ Feature Pass/Fail criteria

5. Sample User Guide

Sample User Guide is based on the original manual of the final outcome. But since we are in the 1st phase of this project development, we cannot describe almost all the things using User Guide.

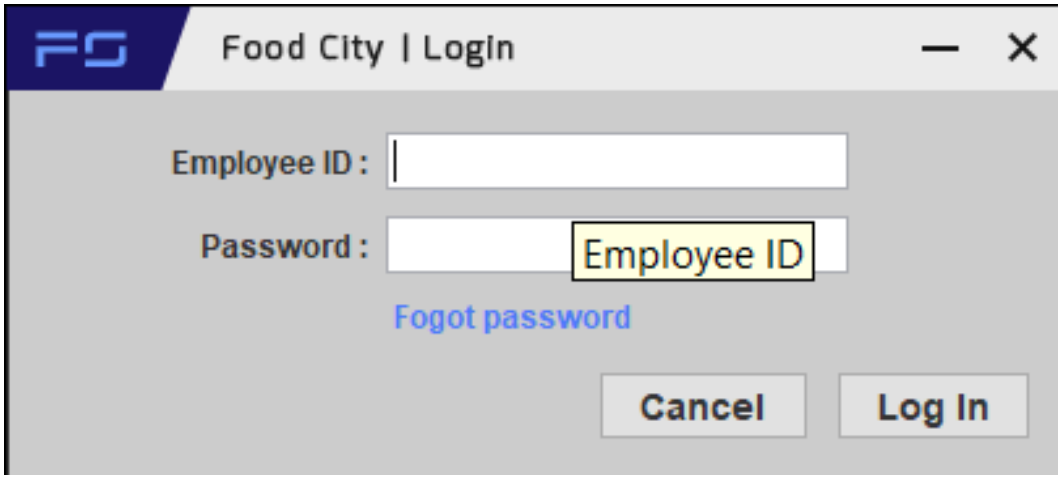
This sample user manual (user guide) is set to help any user who uses the food city System database. In this user manual processors are categorized into several topics and are explained to allow users to understand how to perform specific tasks in a step by step process in the database. Each step describes how the action should be carried out to prevent any errors occurring in the database and have been described with screen shots of pictures of the screen that relates each process.

5.1 System Startup

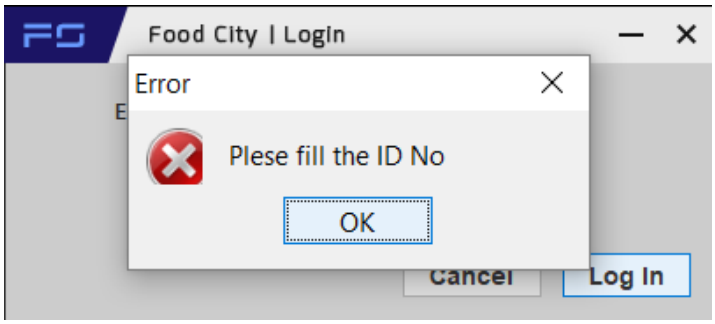
The system is starting with respond following the standard HCI concepts. Following screenshot shows how you see at the software once you click the icon.

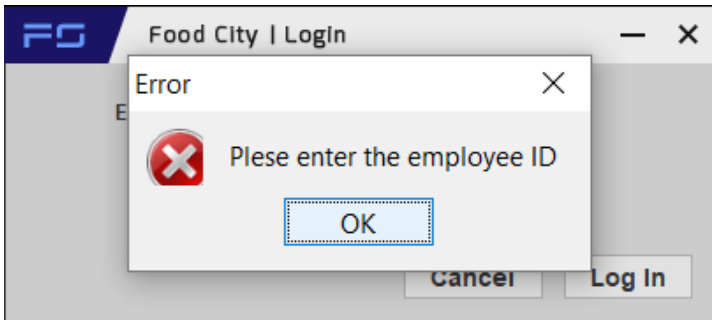
5.2 How to Login to the System

You should enter your user ID and password. You cannot create an ID. It is generated by the system. Also, the default password *(123*) may has to be changed at the first login.



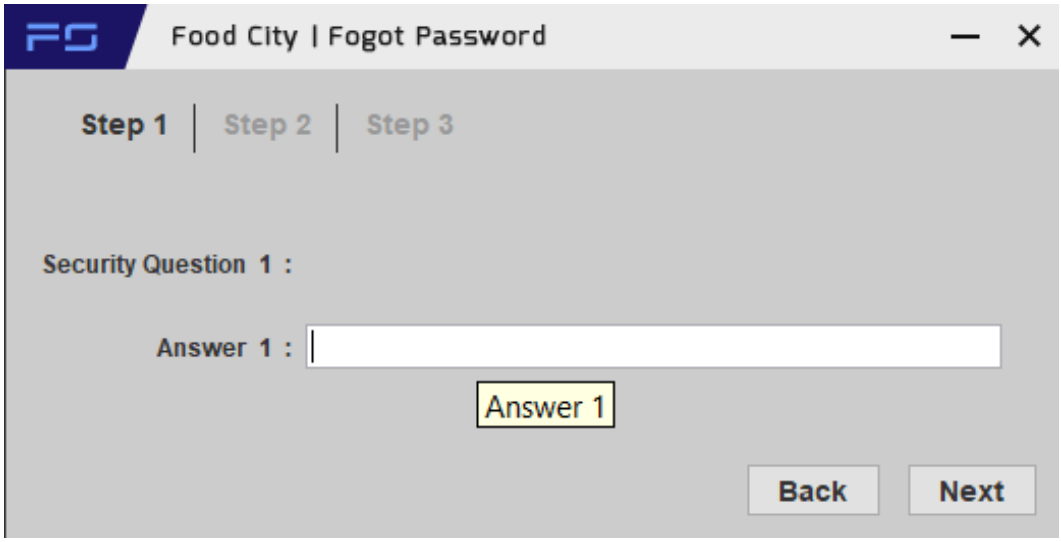
If you missed any field, you may have to enter them correctly. Unless you will see an error message as following screenshot.





5.3 How to Recover Your Password

If you have forgotten your password, please navigate to the “Forgot Password” at the right corner bottom of the login screen and answer the security questions you have added.

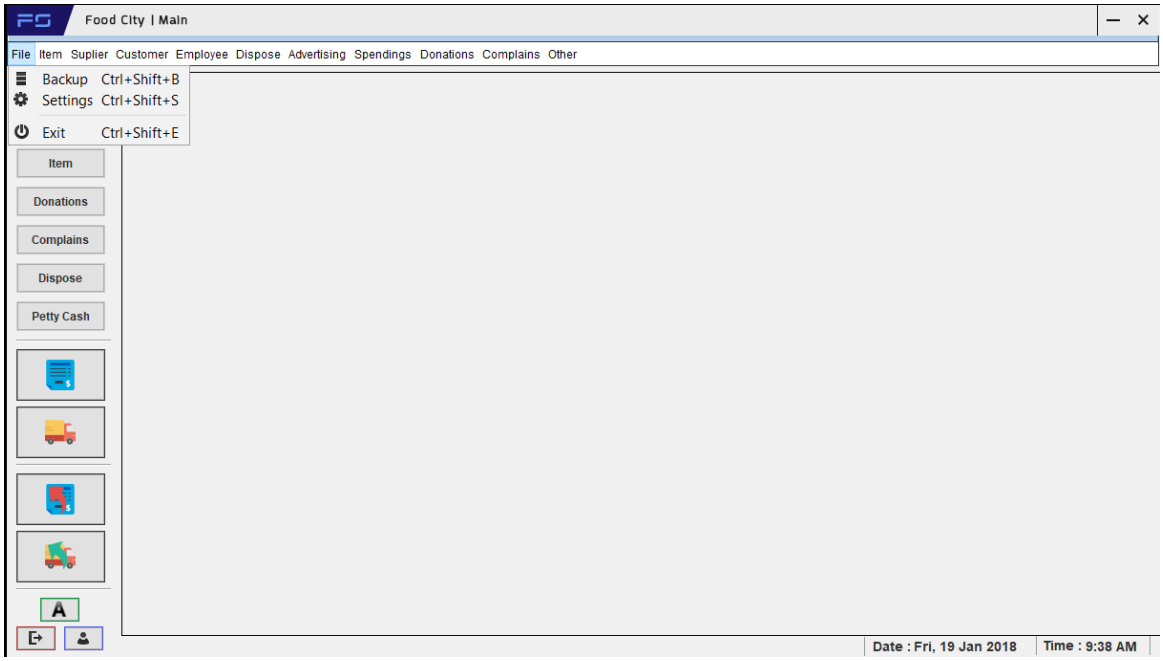


You can proceed using “Next” and if you made any mistake, you can go back in any time at this prompt.

Once you passed all the security checks, you can enter your new password and re-login.

5.4 Main Menu & Switching between Tools

The “Main Menu” has been as the main switch board of the system. The following screen shows the “Main Menu” of the food city system. Through this “Main Menu” user can access to any of the data sources in the system. If the user wishes to exit the system, then user has to click on the “Logout” button.



Please be careful at this screen, Admin Panel is disabled if the user logged in using employee ID.

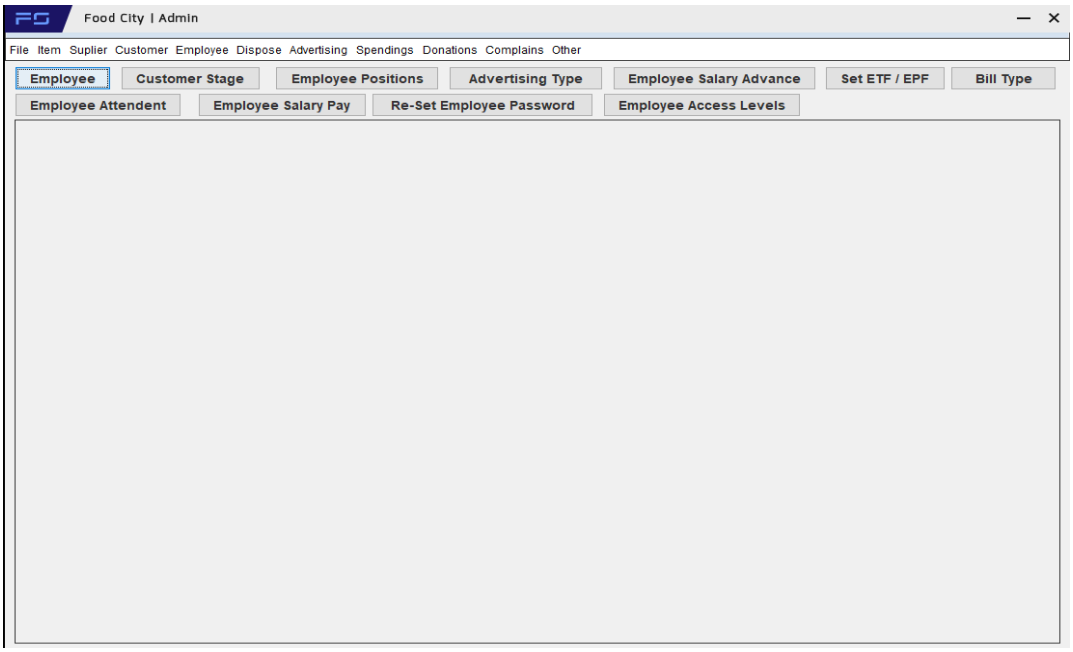
Admin Panel can be accessed using capital “A” at the left bottom of this interface.

Through the “Main Menu” user can view Data Entry Forms and Reports. This system has 2 main roles of users as follows:

- **Admin**
- **Employee**

5.5 Admin Panel

Admin Panel can only be accessed using admin user ID. After you login to the system, if you have admin privileges, you can access to this dashboard clicking "A" button at the left bottom corner.



Admin Panel is consisting with maximum privileges and easily can be navigated clicking buttons on requirement.

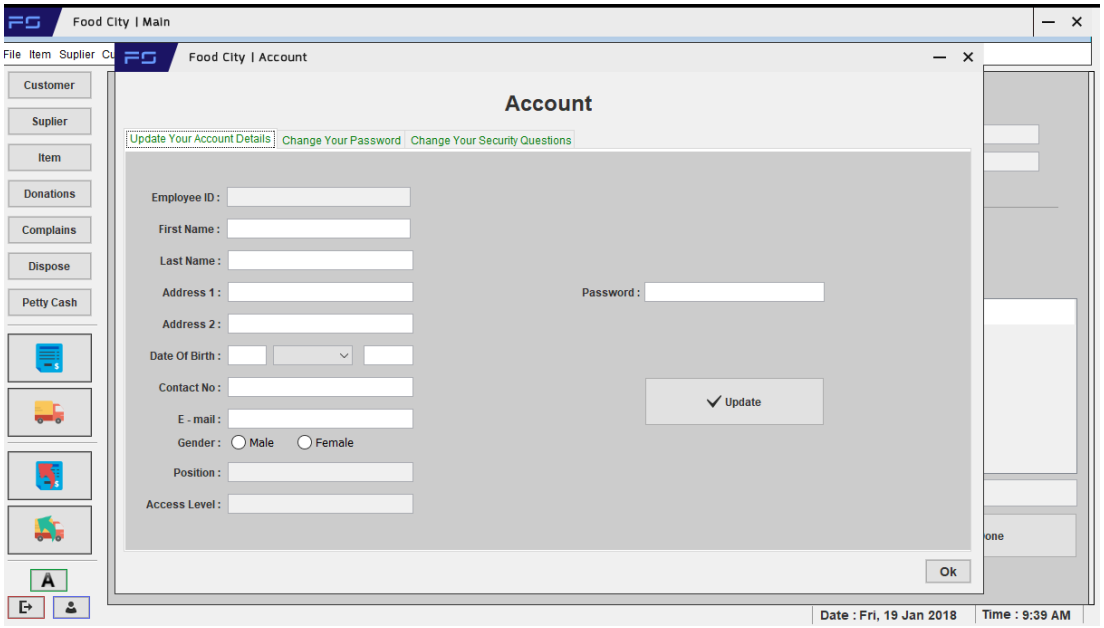
The actions you can take clicking each button has been sorted out here.

- **"Employee" button:** You get the details of employees. admin can employee add or remove.
- **"Customer Stages" button:** Can get the customer details. Admin can add or remove details.
- **"Employee Position" button:** Admin can change the employee position and search the all employee positions.
- **"Advertising Type" button:** Admin can do the advertising for shop and her can show the details of another advertisings.
- **"Employee Salary Advance" button:** Admin want show the details of salary advance her can click this button.
- **"Set ETF/EPF" button:** This button can show the any details of employees' ETF and EPF.
- **"Bill Type" button:** customer get the things in shop. this button is details of bill type.

- **“Employee Attends” button:** Admin pay the payments for employees attends show this button.
- **“Employee Salary Pay” button:** Admin get the information’s for the salary payments.
- **“Re-set Employee password” button:** Sometimes employee forget the password. this button can change the password.
- **“Employee Access Level” button:** Admin can show the employees access panels.

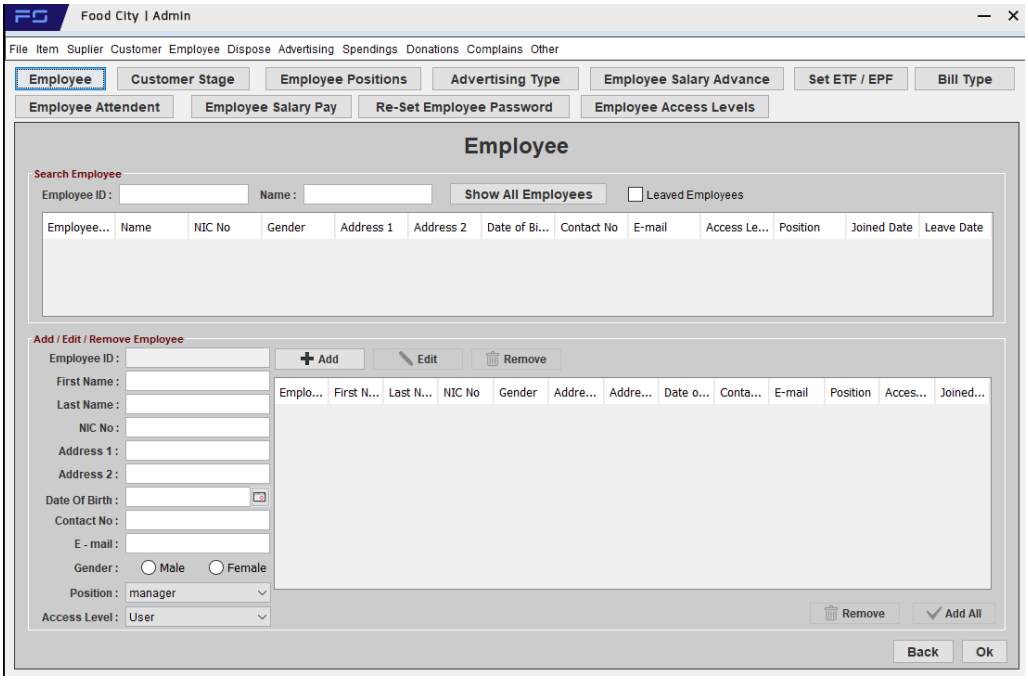
5.6 Admin Panel – Register New Employee

The Admin can add new employee user to the system through following interface. This privilege is only available for Admin user.



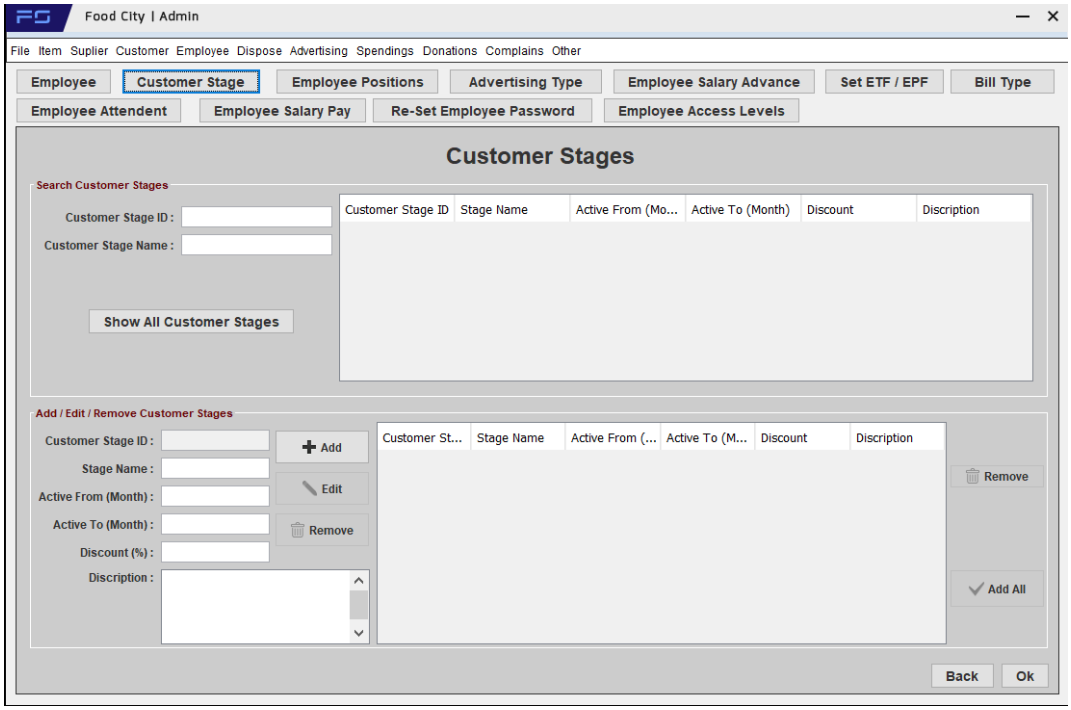
5.7 Admin Panel - Employee

The Admin can add the employee or remove employee. enter the employee id admin can retrieve the details of the employee via following interface.



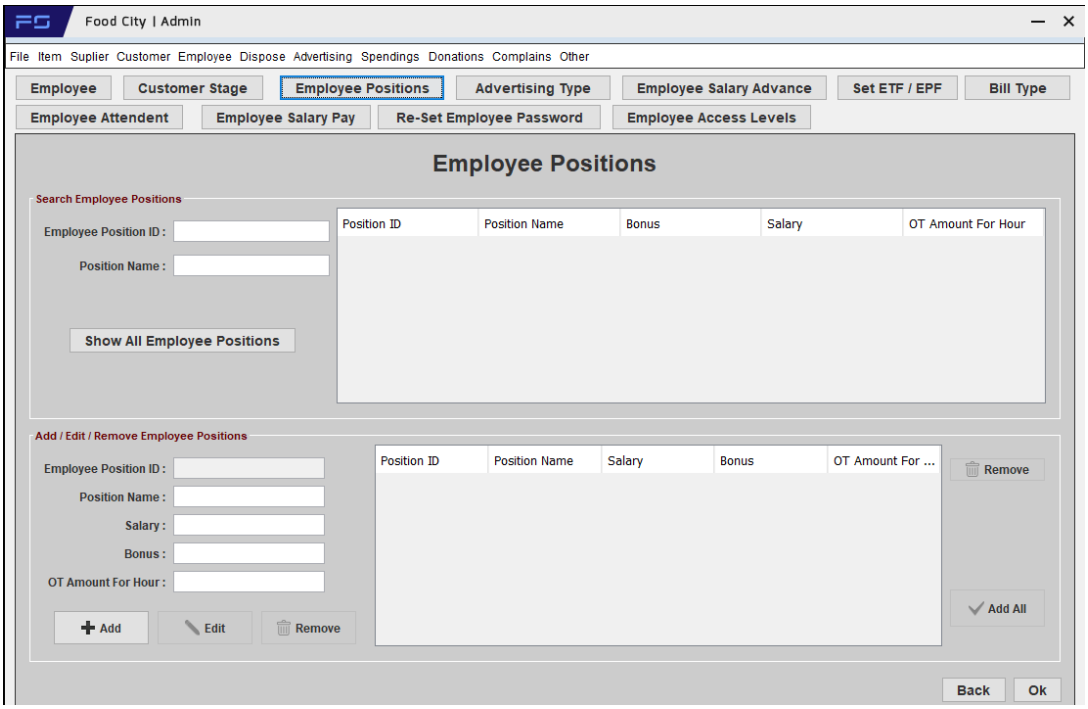
5.8 Admin Panel – Customer Stage

In this panel, Admin can show the customer details add how may customers come the shop. Can search the customer by ID or customer name. It shows all the details of customer. Some customers can be added as Loyalty Customers in this panel. The Admin can remove detail, add detail and edit detail.



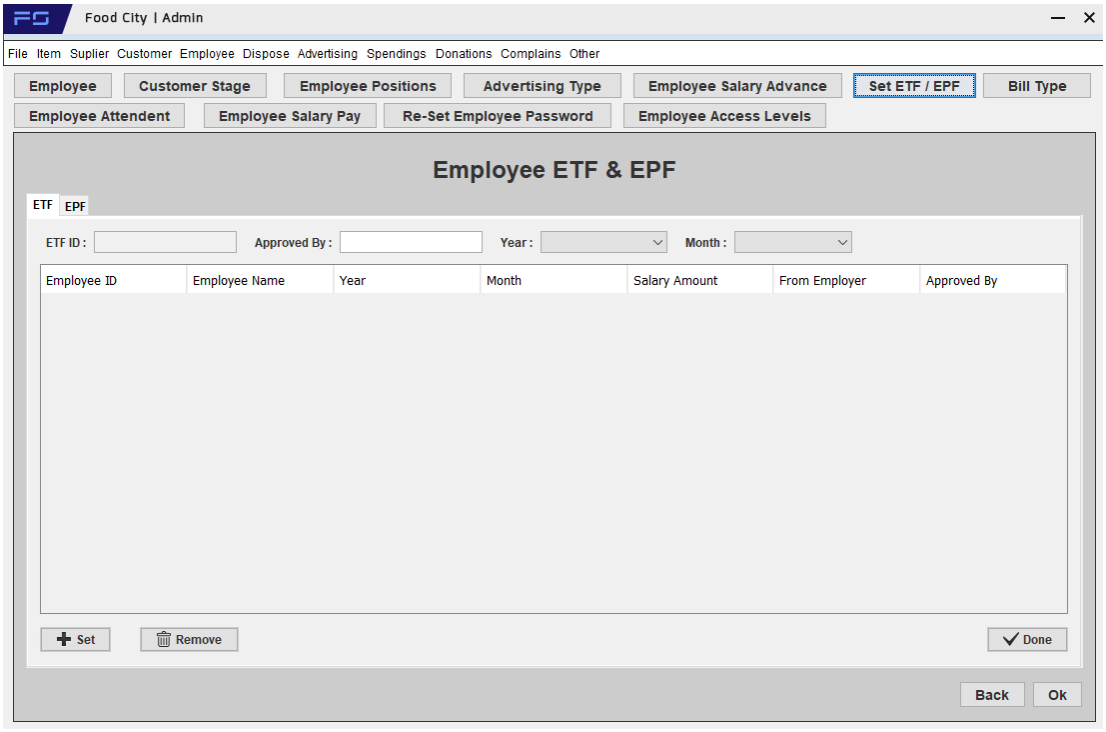
5.9 Admin Panel – Employee Position

Through this interface the Admin User can set the employees' position and change the existing details, add or remove employees.



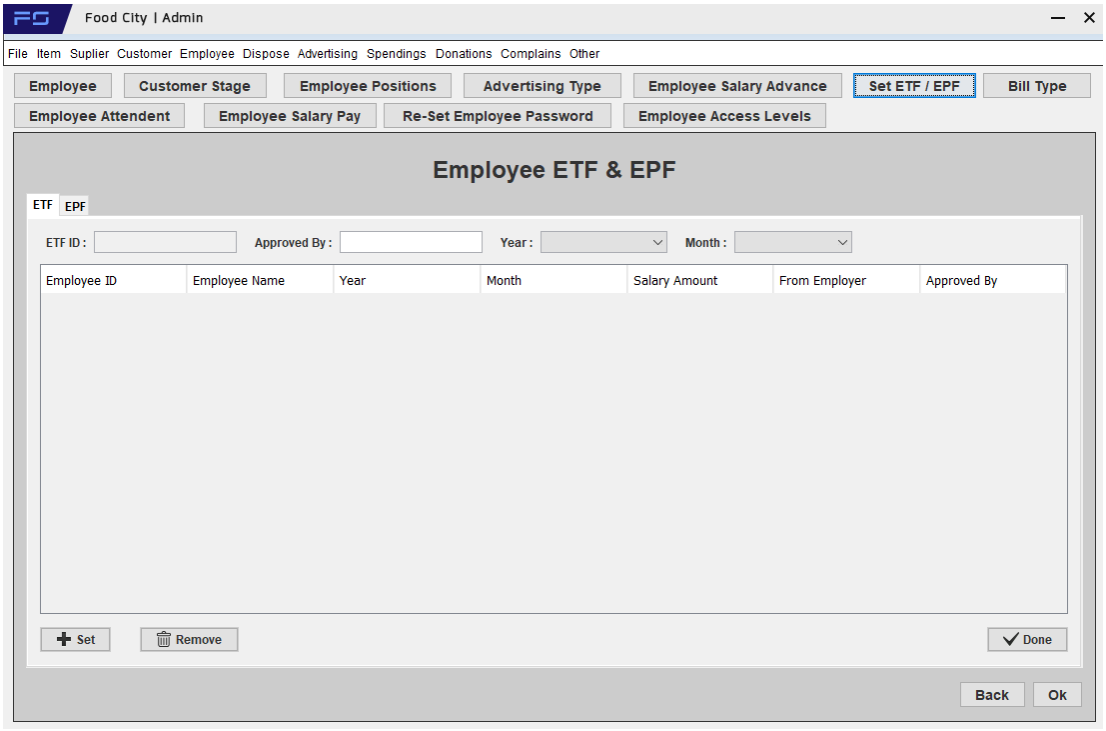
5.10 Admin Panel – Employee Set EPF/ETF

Through this interface the Admin User can set the employees' position and change the existing details, add or remove employees.



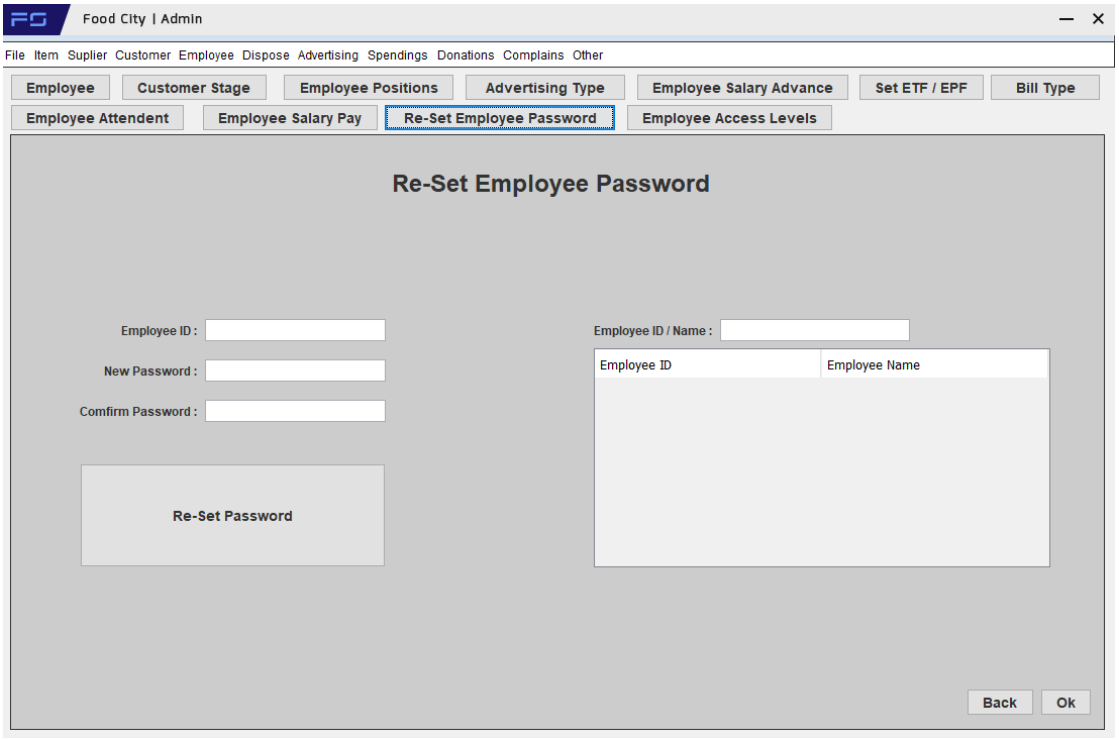
5.11 Admin Panel – Employee Set EPF/ETF

Admin can set ETF & EPF values here.



5.12 Admin Panel – Employee Re-Set Employee Password

Admin is allowed to re-set the passwords of each employee through this interface. To change password, just Admin should know the employee ID followed by new password.

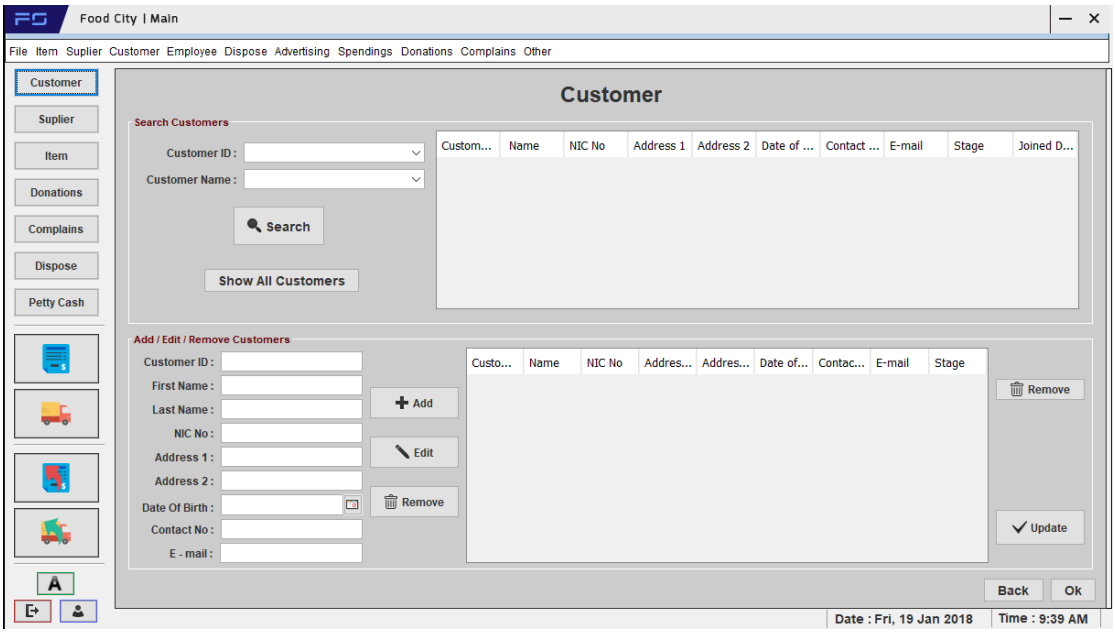


5.13 *Employee Panel*

Employee Panel is the main dashboard which is available for employees. In this panel, users are not allowed to access Admin Panel.

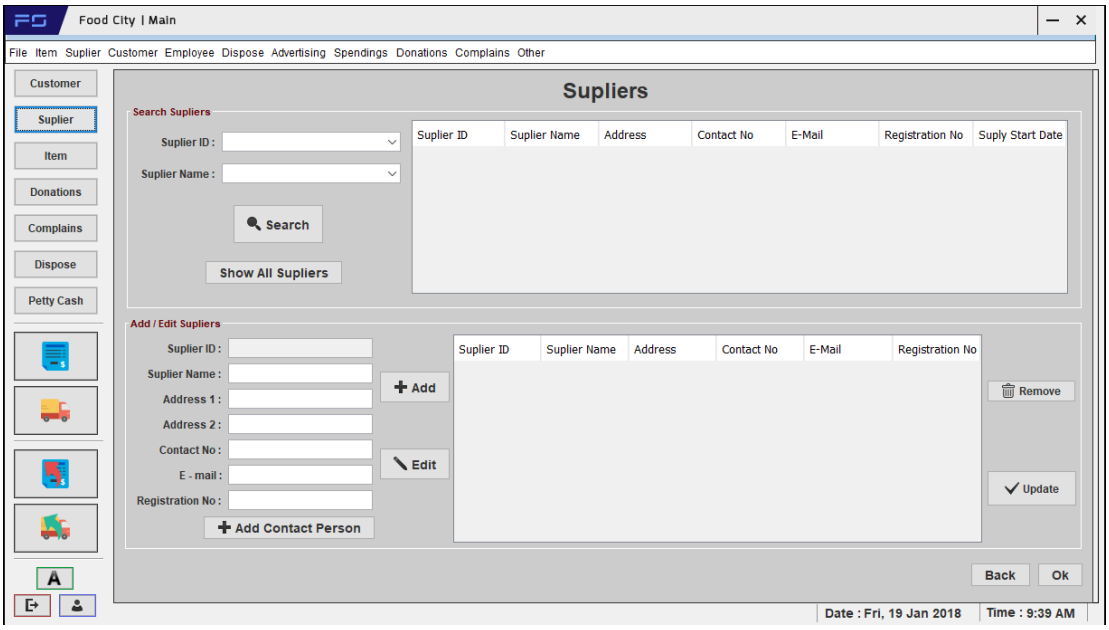
5.14 *Employee Panel - Customer*

Privileged employees can add, change, update, delete customers here. Loyalty based customers only should be added to the system. Each customer can be uniquely identified by the supplier ID. It's generated by the system automatically.



5.15 Employee Panel - Supplier

Through this interface, employee can add, update or remove with all the details of product suppliers of the Food City network. Each supplier can be uniquely identified by the supplier ID. It's generated by the system automatically.



Above screen shows the "Supplier details entry screen" where details of item suppliers can be found.

User can add a new supplier to the system by clicking on the "add new person" button.

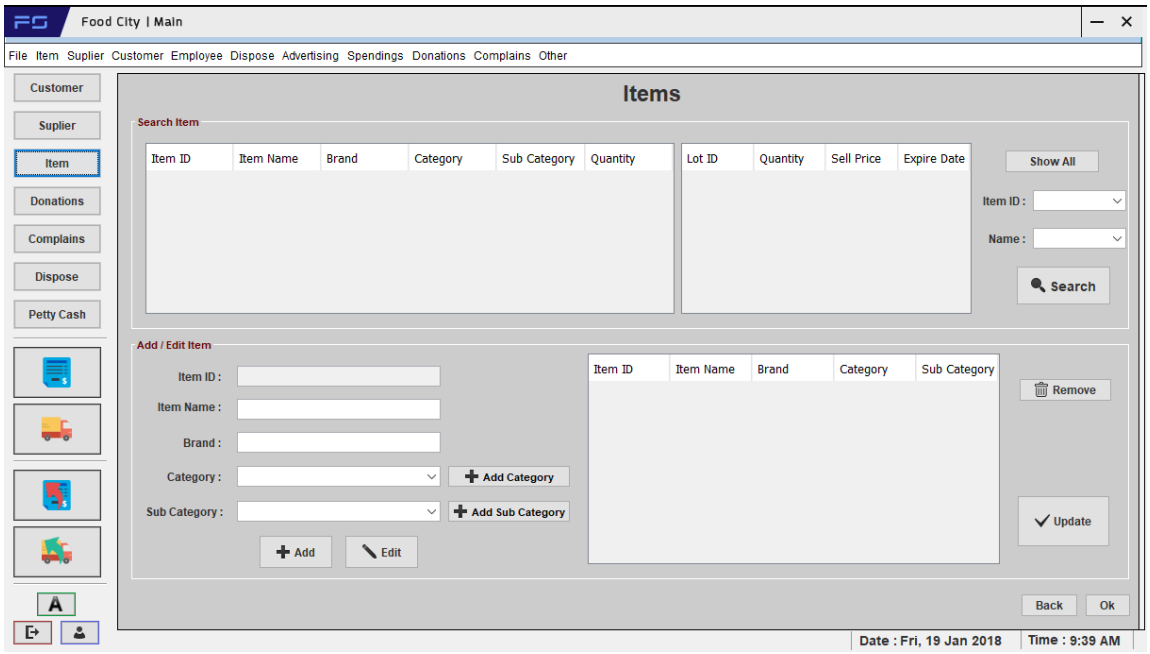
When clicked a confirmation message is displayed on the screen as shown below.

User can search the early supplier enter suppler id ad suppler name by clicking the "search" button.

See all suppler details clicking the "Show AII Suppliers" button.

5.16 Employee Panel - Items

Through this interface, employee can add, update or remove with all the details of products of the Food City network. Each item can be uniquely identified by the Item ID. It's generated by the system automatically.



When user clicks on the "Item Details" button in the main menu under the reports section the below report is displayed on the screen showing the company item details.

The user can add the items category click "Add Category".

User can add sub categories click "Add Sub Category".

Once the user clicks "Add" they can add the items.

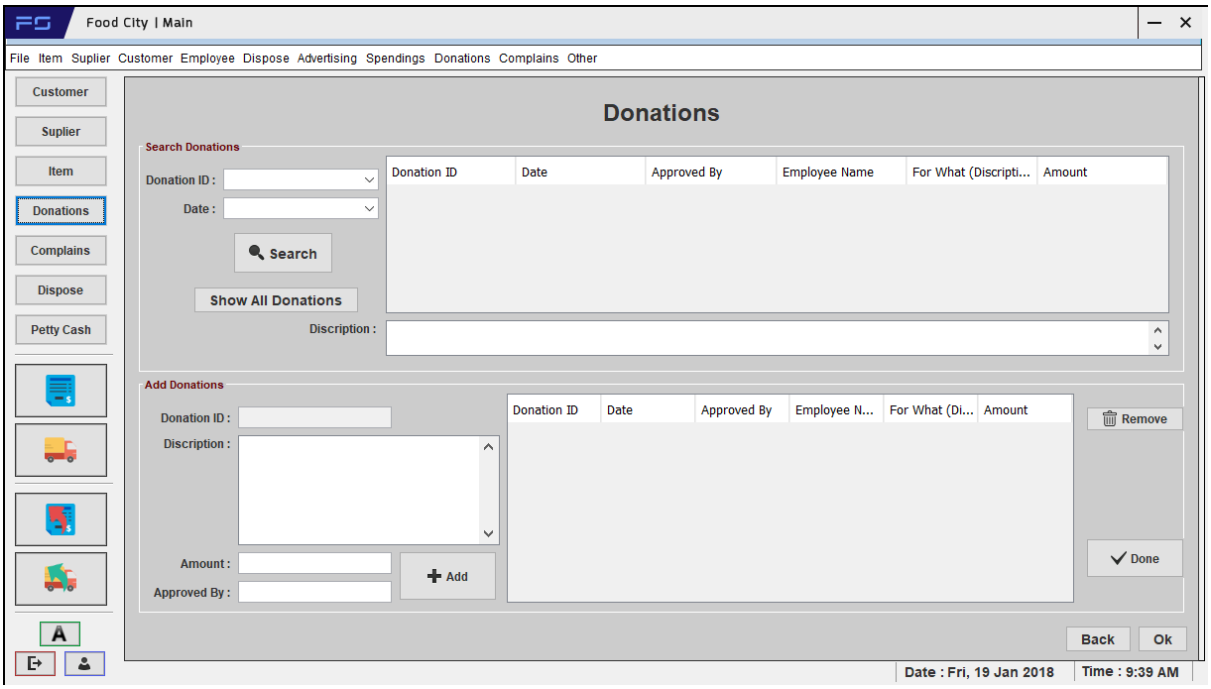
If the "Edit" button is clicked, the user can edit the items.

Enter the item name add item ID user can show the selected item.

User can update the details by click "update" button.

5.17 Employee Panel - Donations

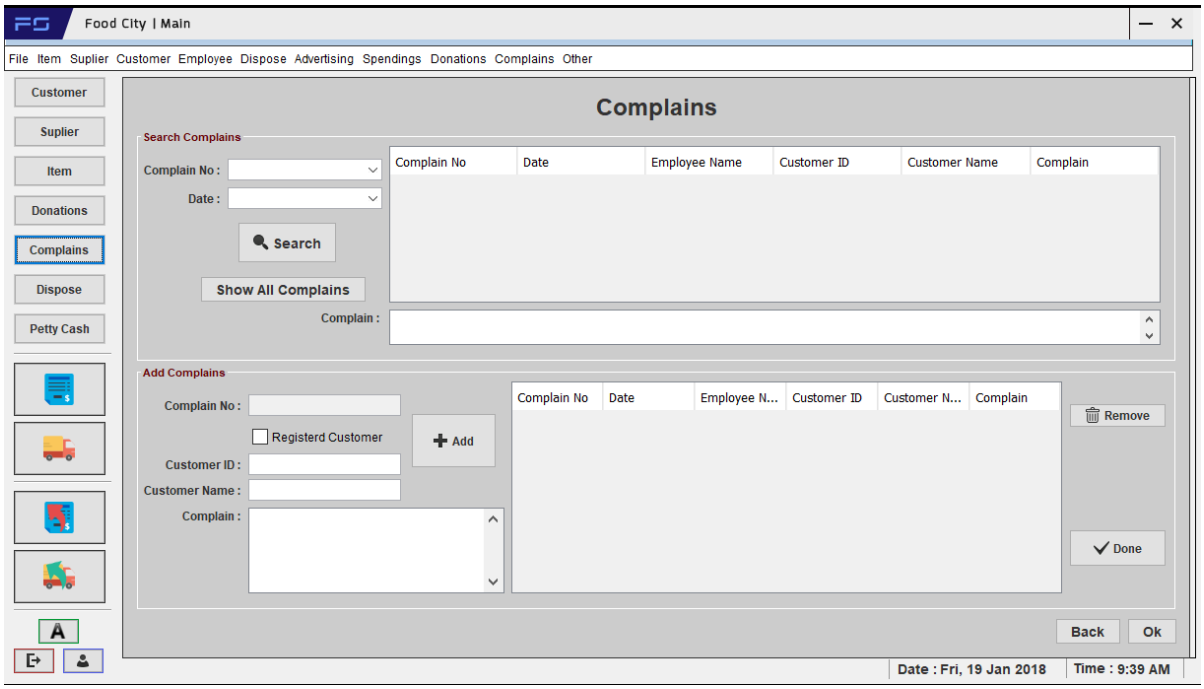
Through this interface, employee can add, update or remove with all the details of donations given by Food City network. Each donation can be uniquely identified by the Donation ID. It's generated by the system automatically.



Through this interface, employee also can search for donations.

5.18 Employee Panel - Complains

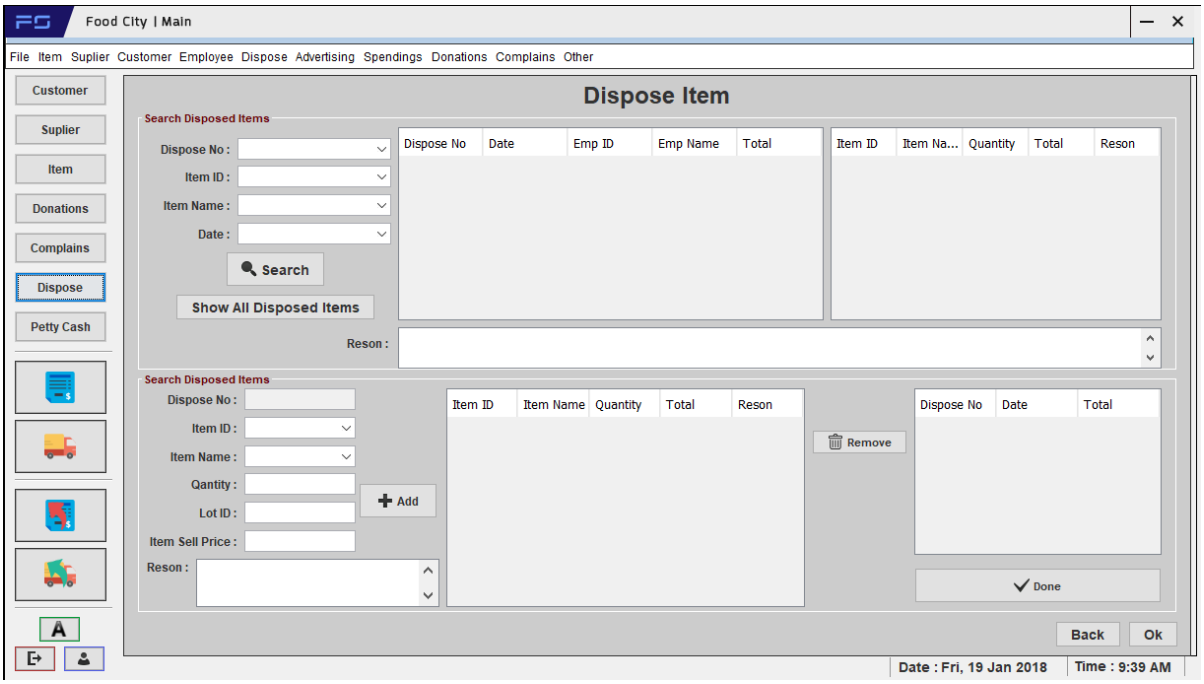
Through this interface, employee can add, update or remove with all the details of complain made by customers. Each complain can be uniquely identified by the Complain ID. It's generated by the system automatically.



Once the "Done" button is clicked, complains can be sent to the database as to show to the administration. 5 or more complains should be listed in the set to be sent.

5.19 Employee Panel - Dispose

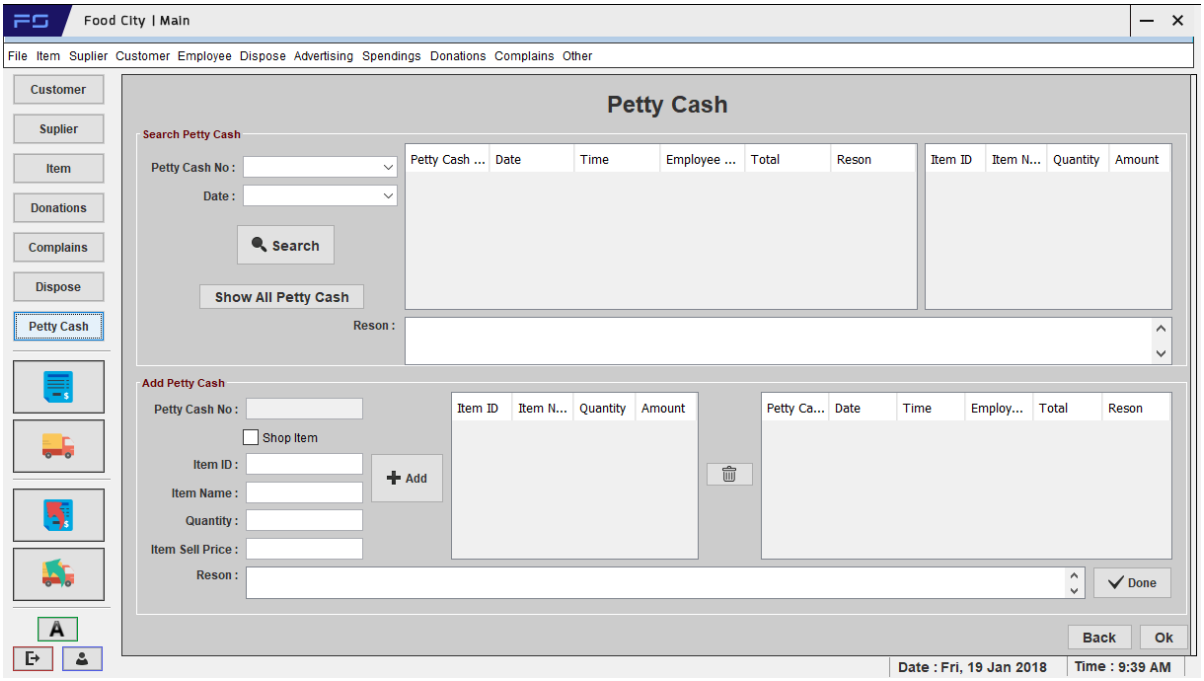
Through this interface, employee can add, update or remove with all the details of aging items or expired items. Each item can be uniquely identified by the Dispose Number referenced to Item ID. It's generated by the system automatically.



This interface allows to generate reports of disposal items of the Food City network.

5.20 Employee Panel – Easy Payments (Petty Cash)

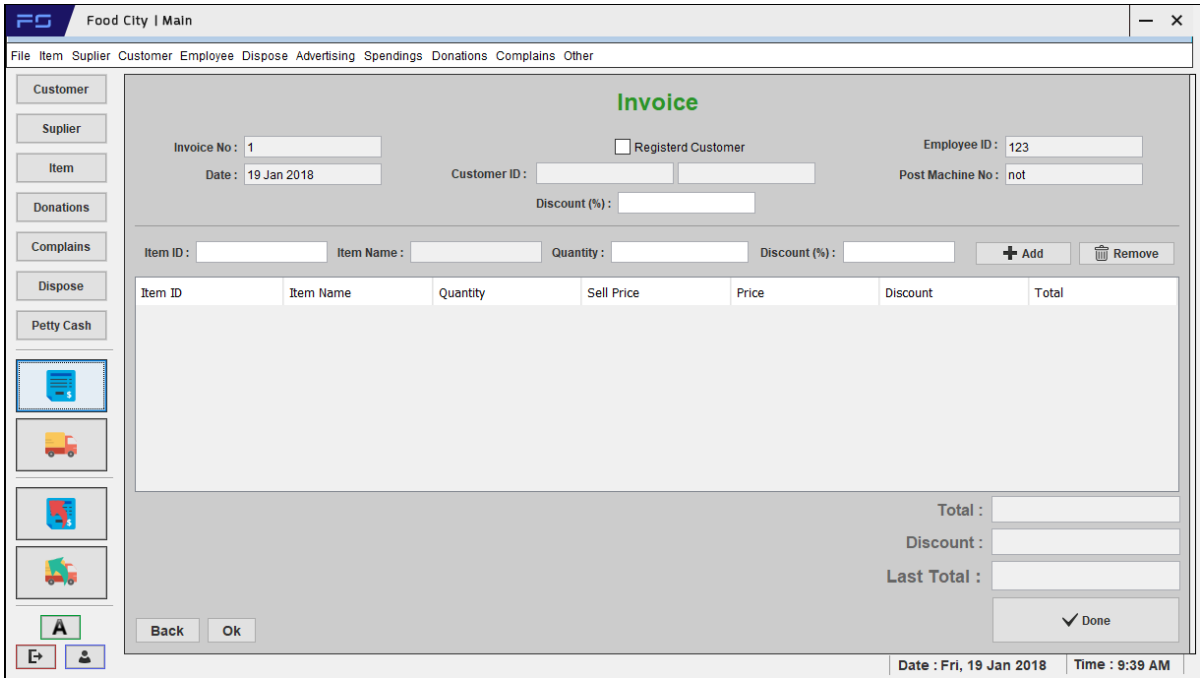
Through this interface, employee can add, update or remove with all the details of aging items or expired items. Each item can be uniquely identified by the Petty Cash Number referenced to Item ID. It's generated by the system automatically.



This interface allows to add customers to pay on easy payment option under credit card option. Easy payments can be searched by Petty Cash ID or Customer ID and "Show All Petty Cash" button would show all the current easy payments to the customers within the Food City network.

5.21 Employee Panel – Invoice

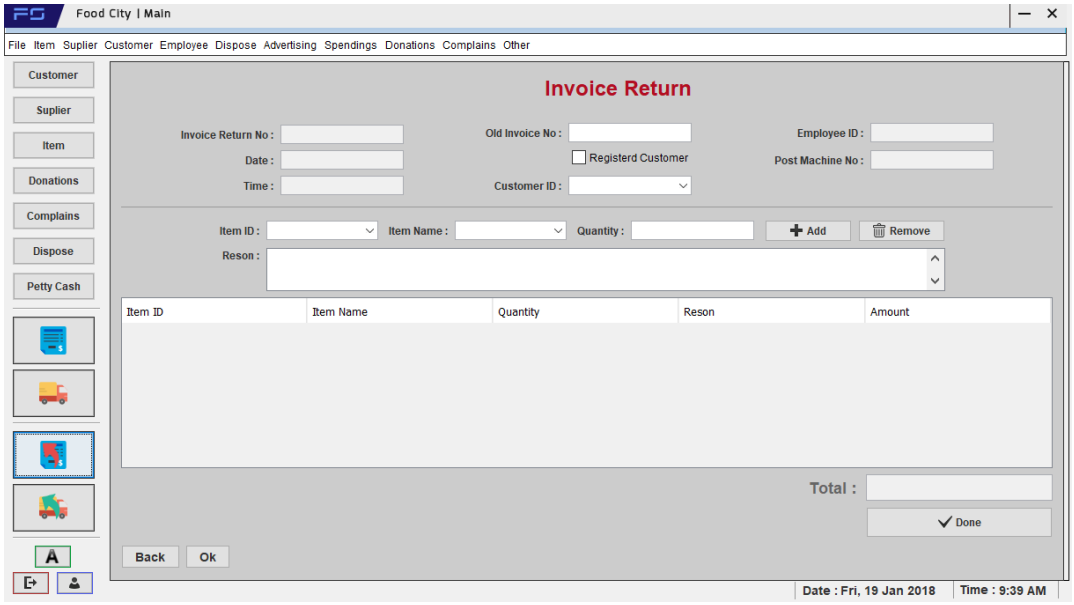
Through this interface, employee can add, update or remove with all the details of aging items or expired items. Each item can be uniquely identified by the Invoice Number referenced to Employee ID. It's generated by the system automatically.



This interface allows to do sales and generate invoices to the bulk of purchased items. If there's a discount for buying items, can also be added here.

5.22 Employee Panel – Return Invoice

Through this interface, employee can add, update or remove with all the details of aging items or expired items. Each item can be uniquely identified by the Invoice Return Number referenced to Old Invoice Number and Employee ID. It's generated by the system automatically.



This interface allows to replace the invoices and return items or refund money to the customers.

6. Conclusion

The main objective of this report was to design a touchscreen interface for a food city system.

At the start of the project we visited the Food City and inspected their day to day processors. Then, we discussed the requirements need to include, sections we need to highlight and conventions we need to come up when designing a user-friendly Food City Management System.

Then as a team we discussed in detailed about those sketches using usability methodologies and selected two designs out of five to proceed to the final designing stage which is designing the interfaces as a prototype. Finally, we conducted a usability evaluation to find out whether the system is efficient and effective based on the user's point of view.

7. Appendixes

8. References

- ***Mini Project Report – Sample Guidance***

http://www.cs.northwestern.edu/~agupta/_projects/embedded_systems/FinalReport.pdf

- ***Design Principals***

http://www.designprinciplesftw.com/collections/shneidermans-eight-golden-rules-of-interface-design

- ***Usability Methodology by Jacob Neilsen***

https://www.nngroup.com/articles/ten-usability-heuristics

- Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, Communications of the ACM 33, 3 (March), 338-348.

- Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April), 249-256.

- Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. Proc. ACM CHI'94 Conf. (Boston, MA, April 24-28), 152-158.

- Nielsen, J. (1994b). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods, John Wiley & Sons, New York, NY

Feedbacks

END OF THE REPORT