

College of Science and Technology

School of Science and Technology

# SOFT30121: Advanced Analysis and Design

# Systems Analysis Design and Implementation

*David Vladimirovas N0899068*

*Hassaan Naveed N0898071*

*Sripriya Vattikula N0894013*

*Eisvinas Pranckaitis N0873702*

*Nikita Khera N0875673*

# Disclaimer

*This report and the program provided to the tutor are the results of the members of the team as indicated below. All sources of information are referenced within the document and/or source code. Any contributions to the work by third parties, other than tutors are stated clearly below this declaration. Should this statement prove to be untrue we recognize the right and duty of the Board of Examiners to take appropriate action in line with the university’s regulations on assessment.*

*ID No: N0899068, N0898071, N0894013, N0873702, N0875673*

Contents

[SOFT30121: Advanced Analysis and Design 1](#_Toc95493585)

[Systems Analysis Design and Implementation 1](#_Toc95493586)

[Disclaimer 2](#_Toc95493587)

[Design Documentation 4](#_Toc95493588)

[Architecture Diagram 4](#_Toc95493589)

[Deployment Diagram 7](#_Toc95493590)

[Process Diagram 8](#_Toc95493591)

[Sequence Diagrams 13](#_Toc95493592)

[Structure Diagrams 20](#_Toc95493593)

[User Manual 21](#_Toc95493594)

[Creating an account and logging in 21](#_Toc95493595)

[Chef and Head chef 21](#_Toc95493596)

[Head chef privileges 21](#_Toc95493597)

[Delivery Personnel 22](#_Toc95493598)

[System admin 23](#_Toc95493599)

[H&S worker 24](#_Toc95493600)

[Acceptance Test Plan 25](#_Toc95493601)

[References 34](#_Toc95493602)

[Appendix 35](#_Toc95493603)

[Appendix 1 35](#_Toc95493604)

# Design Documentation

## Architecture Diagram

### Application Structure

The application ‘Smart fridge’ will utilise the client-server architecture in the attempt to balance the processing between the client and server. The general queries and data will be stored on the server, allowing for quick and easy access to required data. Due to the usage of the application, the client’s device will vary in hardware, making it difficult to determine whether the device will be able to maintain constant performance on that device. To temporarily solve this issue, the architecture used will allow for the processing to be split between the client’s device and the server.

This implementation will bring forth the following benefits:

**Data sharing –** Due to the centralized storage of data, monitoring and manipulating the data being communicated is easy and quick, especially with the usage of SQL queries, making efficient use of data manipulation techniques.

**Service integrations –** Integrating a service will become accessible by the clients that are connected the server

**Data processing –** Data can be processed despite the location as there will be access to the server, where either the client’s device or server will process the data.

**Maintenance –** Due to the server being run on a single virtualized machine, it makes it easier to upgrade or repair the machine. This means that the general cost will be lower and will generally be quick.

Whilst there are benefits, there are disadvantages that need to be accounted for, of which are:

**Server overloads –** If the number of clients were to use the application, the server will inevitable overload due to the traffic congestion. This is a careful consideration that was required to take as it would greatly affect the usage of the application, where the delivery drivers may cause the phantom effect if the application was to perform very slowly and delay the drivers.

**Negatives of centralized architectures –** It is a given that having a centralised architecture will result in unfavourable situations. The most common and costly issue would be that is the centralized server were to fail, the clients request would not be completed.

### Architecture Pattern

For the architectural pattern of the application, it was decided that using a model-view-controller (MVC) pattern would be most suitable. This is because of many reasons, one of which, in abstract terms, is that the application will be fetching data from a server and presenting that data in a structured manner to a user. As the architectural patter is supported heavily in Java, it makes it easier to test, and expand the application since the application is also programmed in java.

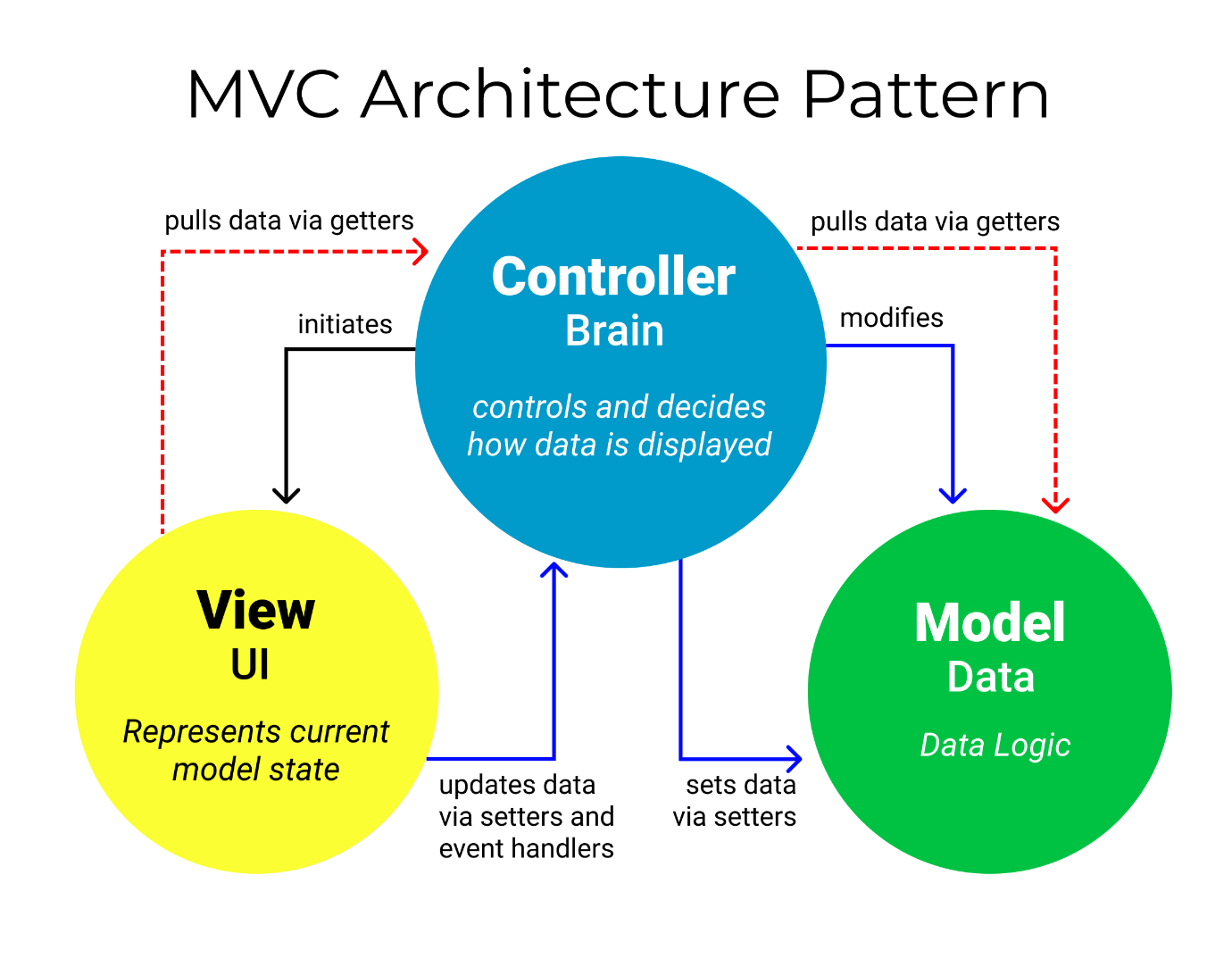


Figure 1: Diagram showing the process of an MVC architectural pattern, that was first introduced by Trygve Reenskaug (Rafael D. Hernandez, 2021)

For a simple explanation, the MVC pattern broken down into 3 different components:

**Model –** Represents data, where is stores and manages data (NOTE: does not depend on the controller or the view). In our application, the ‘model’ would be the database server that would store all the client information and data for each fridge. The database server that is being used is ‘Firebase’, where data will be stored in a JSON format and synchronized to each connected client.

**View –** Graphical user interface. In context of our application, this would be the representation of data, the UI. The user would perform actions on the GUI such as pressing a button or inputting data into a text field.

**Controller –** Provides the model data to the view, interpreting user actions such as UI interactions. The controller will be our event handler in the application where will process and validate with the ‘model’ to return information to the ‘view’, for display to the user again.

Benefits:

* Code maintenance is easy, making it easy to maintain and grow
* The MVC model can be test independent from the user, making it easier to implement changes. Additionally, parallel development can be performed on various components
* Reduction in complexity as the components are divided, which makes it easier and quicker to implement.
* Allows for support for test-driven development
* Compatible with web application as it is supported heavily by large teams of web designers and developers
* As all classes and objects are independent, it is easier to test them individually.

Negatives:

* The code can become difficult to read over time and reuse
* The ‘controller’ requires lots of maintenance as it will be updated the most
* Knowledge of several technologies will be required to achieve peak performance and efficiency
* The data communication will become complex and inefficient over time as it grows unless the application expands and utilizes external services.

The following diagram will show the general architecture of the application:

Diagram

Description automatically generated

## Deployment Diagram

Deployment diagrams are a type of UML (Unified Modelling Language) diagrams that represent hardware and software components that are needed for the application, where they help with visualising the topology of the system.

A picture containing text, sky, screenshot

Description automatically generated

Figure 2: Diagram representing the physical topology of the application.

The diagram above shows the deployment of the developed mobile application. “Smart Fridge” mobile application is launched on an android smartphone, where it connects to the server database when a client logs in. The application has several controllers, which are responsible for their respective functionalities, retrieving the user input and managing the Model view. The Model is responsible for updating the GUI that is shown to the user and retrieving/sending information from/to the database, which contains several tables needed for the application.

## Process Diagram

Process diagrams, also known as “flow diagrams”, are diagrams that model sequence/s of activities in a process (Phillipe D., Gilbert Raymond, 2014). The process diagrams will hold information about user intents, participants, what triggers events and more, thus, making it easier to visualise the ‘process’ the application will adhere.

Use case diagrams are a graphical representation of the possible interactions/paths a user can take within the system.

### Use Case 1

Diagram

Description automatically generated

Figure 3: Use case diagram for creating a new account

|  |  |
| --- | --- |
| **Use Case Name:** Login and Creating a new account. | **Importance Level:** Moderate |
| **Primary Actor:** User | |
| **Brief Description:** Shows how the user were to proceed in making a new account. | |
| **Trigger:** User clicks the create account button on main page  **Trigger Type:** External | |
| **Relationships:**  Association:  Include: Enter user into the Database, Alert user about wrong format of input  Extend:  Generalization: | |
| **Normal Flow of Events:** User enters credentials and creates an account | |
| **Sub: Flows:** N/A | |
| **Alternative/Exceptional Flows:** User may be prompted to enter credentials again | |

### Use Case 2

|  |  |
| --- | --- |
| **Use Case Name:** Login / Log Out | **Importance Level:** High |
| **Primary Actor:** User | |
| **Brief Description:** This is where the user logs into the system and the system has to validate the users’ details by communicating with the database. | |
| **Trigger:** User enters details and presses login.  **Trigger Type:** User | |
| **Relationships:**  Association: Communicate with server software.  Include: Application closes, validate credentials, compare credentials.  Extend: Press logout, load user profile, enter username/password, login  Generalization: | |
| **Normal Flow of Events:** User enters details and logs in. | |
| **Sub: Flows:** User log out. | |
| **Alternative/Exceptional Flows:** User is denied access. | |

Diagram

Description automatically generated

Figure 4: Use Case diagram for logging in and logging out

### Use Case 3

|  |  |
| --- | --- |
| **Use Case Name:** Head Chef | **Importance Level:** High |
| **Primary Actor:** Head Chef | |
| **Brief Description:** This is where the head chef logs into the system exercises his power of authorization. | |
| **Trigger:** Head Chef enters details and presses login.  **Trigger Type:** User | |
| **Relationships:**  Association: Communicate with server software.  Include: Enter Product quantity, search stock level, grant users’ access to fridge, allow users to insert/remove food items  Generalization: | |
| **Normal Flow of Events:** Head Chef logs into his account using his credentials. He can insert/remove the food items in the fridge by entering the product, quantity. Search for the stock levels and exercise his authority to grant users access to fridge and allow users to insert/remove food items. | |
| **Sub: Flows:** User may use the search bar to search for the stock etc. | |
| **Alternative/Exceptional Flows:** User is denied access. | |

Diagram

Description automatically generated

Figure 5: Use Case diagram for Head Chef interactions

### Use Case 4

|  |  |
| --- | --- |
| **Use Case Name:** Chef | **Importance Level:** High |
| **Primary Actor:** Chef | |
| **Brief Description:** This is where the chef logs into the system to insert/remove the food items and search the stock levels. | |
| **Trigger:** Chef enters details and presses login.  **Trigger Type:** User | |
| **Relationships:**  Association: Communicate with server software.  Include: : Enter Product quantity, search stock level  Extend:  Generalization: | |
| **Normal Flow of Events:** Chef logs into his account using his credentials. He can insert/remove the food items in the fridge by entering the product, quantity, and also Search for the stock levels | |
| **Sub: Flows:** User may use the search bar to search for the stock etc. | |
| **Alternative/Exceptional Flows:** User is denied access. | |

Diagram

Description automatically generated

Figure 6: Use Case diagram for the Chef user role interactions

### Use Case 5

|  |  |
| --- | --- |
| **Use Case Name:** Delivery Guy | **Importance Level:** High |
| **Primary Actor:** Delivery Guy | |
| **Brief Description:** This is where the delivery guy logs into the system to insert the food items and look for items replenished. | |
| **Trigger:** Delivery Guy enters details and presses login.  **Trigger Type:** User | |
| **Relationships:**  Association: Communicate with server software.  Include: Enter Product quantity and units  Extend:  Generalization: | |
| **Normal Flow of Events:** Delivery Guy logs into his account using his credentials. He can insert the food items in the fridge by entering the product, quantity, and also look for the replenished food items. | |
| **Sub: Flows:** User may use the search bar when looking for the replenished food items | |
| **Alternative/Exceptional Flows:** User is denied access. | |

Diagram

Description automatically generated

Figure 7: Use Case diagram for the user role Delivery guy

## Sequence Diagrams

Lucid Charts state that sequence diagrams are diagrams that describe how and in what order a group of objects work together to understand requirements for a system or document and existing process (2021).

### Diagram Description automatically generatedCreating an account

Figure 8: Sequence diagram for creating an account

### Logging in

Chart, bar chart

Description automatically generated

Figure 9: Sequence diagram for logging in

### Head Chef

Diagram

Description automatically generated

Table

Description automatically generated with medium confidence

Diagram, table

Description automatically generated

Diagram

Description automatically generated

Figure 10: Sequence diagram for the head chef

### Chef

**Diagram

Description automatically generated**

**Diagram

Description automatically generated**

**Table

Description automatically generated**

Figure 11: Sequence diagram for the chef

### Delivery Person

Table

Description automatically generatedTable

Description automatically generated with low confidence

Diagram

Description automatically generated

**Diagram

Description automatically generated**

Figure 12: Sequence diagram for the delivery person

## Structure Diagrams

As the name implies, theses diagrams refer to the structure of a program or application. Structure diagrams show a graph of the structure of the program, including the elements that compose the program and their relationship (IBM, 2016).

### Component Diagram

The general purpose of a component diagram is to show the relationships between components of the system. These components can be referred to as classes representing a system or subsystem (Lucid Software Inc, 2022)

*Refer to appendix 1.*

### Class Diagram

One of the recommended and used diagrams for creating an application is the class diagram. This diagram describes what must be present in the system being modelled. To represent class diagrams, UML is often used as show in the following figure.

Diagram

Description automatically generated with low confidence

Figure 13: UML Class diagram

# User Manual

## Creating an account and logging in

**How to register an Account?**

1. In the *Login* window press **Create Account** button.
2. In the *Create Account* window enter a **Username**, **Password**. After that choose the **Account Type** you want to register as.
3. After filling in all the required information and choosing the account type, press the **Create Account** button.

**How to login into an already existing account?**

1. First, login into your account. To do that, in the *Login* window press **Staff Login** button.
2. Enter your account credentials in the appropriate Username and Password text fields and press **Login** button.

## Chef and Head chef

Chefs and head chefs can check what items are in the fridge, they can insert and remove items from the fridge as well.

**How to see the contents of the fridge?**

1. First, login into your account.
2. In the *Home* window press the **View Items** button, to see the contents of the fridge.

**How to insert an item into the fridge?**

1. First, login into your account.
2. In the *Home* window, press the **Add Item** button.
3. Enter the information in the appropriate text fields about the product you want to insert in the fridge: Name of the item, it’s Expiry Date and Quantity.
4. Press the **Add Item** button to put the item in the fridge.

**How to remove an item from the fridge?**

1. First, login into your account.
2. In the *Home* window, press the **Remove Item** button.
3. Enter the information in the appropriate text fields about the product you want to take out of the fridge: Name of the item and Quantity.
4. Press the **Remove** button to take the item out of the fridge.

## Head chef privileges

Head chefs have a bit more privileges than the chefs. They can set a threshold for a product, so they would know when it is running out. Head chefs are also able to manage fridge access, confirm product orders and receive notifications about items that passed their threshold or are about to expire.

**How to set a threshold for an item?**

1. First, login into your account.
2. In the *Home* window press the **Manage Stock** button.
3. Enter the Name of the item and the Restocking Threshold in their appropriate text fields.
4. Press the **Set** button to set a threshold.

**How to manage fridge access?**

1. First, login into your account.
2. In the *Home* window press the **Manage Users** button.
3. Enter the Username of the person you want to change fridge privileges to.
4. Select appropriate Fridge Access switch - green means allow, grey means deny.
5. Press the **Submit** button.

**How to make an item order?**

1. First, login into your account.
2. In the *Home* window press the **Confirm Order** button.
3. A list will appear in *Complete Order* window, where you will be able to see what items will be ordered.
4. Press the **Confirm** button to order products.

**How to see product notifications?**

1. First, login into your account.
2. In the *Home* window press the **Notifications** button.
3. In the *Notifications* window a list will appear, containing information about items that are about to expire or run out.

## Delivery Personnel

Delivery personnel can only insert items in the fridge in order to complete their order.

**How to complete an order?**

1. In the *Login* window press the **Delivery Login** button.
2. Enter the Pin and press the **Login** button.
3. In the appropriate text field provide the Name of the item, its Expiry Date and Quantity.
4. Press the **Add Item** button to add the item into the fridge. This process of steps 3 and 4 will have to be repeated for every item on the order;
5. After you added all the items from the order, press the **Complete Delivery** button.

## System admin

A system admin is able to see all the registered user accounts, manage fridge access, delete a user account, promote or demote head chef in the system and change the pin to the fridge.

**How to see all registered users?**

1. First, login into your account.
2. In the *Home* window, press the **View Users** button.
3. In the *View Users* window a list will appear containing all user accounts.

**How to manage fridge access?**

1. First, login into your account.
2. In the *Home* window press the Manage Users button.
3. Enter the Username of the person you want to change fridge privileges to.
4. Select appropriate Fridge Access switch - green means allow, grey means deny.
5. Press the **Submit** button.

**How to delete a user?**

1. First, login into your account.
2. In the *Home* window, press the **Delete Users** button.
3. Enter the Username of the person, whose account you want to delete.
4. Press the **Delete** button.

**How to promote a chef to head chef?**

1. First, login into your account.
2. In the *Home* window, press the **Change Head Chef** button.
3. Enter the Username of the chef you want to promote to head chef.
4. Press the **Promote** button.

**How to demote a head chef to chef?**

1. First, login into your account.
2. In the *Home* window, press the **Change Head Chef** button.
3. Enter the Username of the chef you want to demote to chef.
4. Press the **Demote** button.

**How to change the fridge pin?**

1. First, login into your account.
2. In the *Home* window, press the **Change Pin** button.
3. Enter the New Pin in the appropriate text field.
4. Press the **Submit** button to change the pin to the fridge.

## H&S worker

Health and safety workers account allows the user to see what items are in the fridge and what is their expiration date.

**How to see the report about the items in the fridge?**

1. First, login into your account.
2. In the *Home* window, press the **View Report** button.
3. In the *View Fridge* window an item list will appear containing all products in the fridge and their expiry dates.

# Acceptance Test Plan

|  |  |  |
| --- | --- | --- |
| **Test** | **Procedure** | **Outcome** |
| Allow the user to register. | The user must be able to register their first name, surname, work title (i.e., chef, delivery driver etc.) and password. | Graphical user interface, application  Description automatically generated  User being able to create an account that can be sys admin, chef or head chef. |
| The user has registered and allowed access to the fridge. | The system will only allow access to the fridge after a user had logged in. | Table  Description automatically generated with medium confidence  The user has logged in successfully and the fridge. |
| The user would have logged out as the fridge is closed. | The system will log out whenever a user has closed the fridge | \*press back button\* |
| Registered user must have assigned work titles within the system. | The system must assign authority to the registered users in correspondence to their work title | Graphical user interface  Description automatically generated with medium confidence  This is an example of the registered system admin who is assigned. There is also chef and head chef. |
| The delivery people have access to the back door fridge and put items into the fridge. | Users under the “delivery person” work title have access to the back door of the fridge and can only insert items into the fridge | Chart, waterfall chart  Description automatically generated  The delivery people have a pin to have access to the back door of the fridge. |
| Testing the system when the fridge door is opened. | The system must have a rule for when a door Is open on either side, the other side cannot be opened until the opened door is closed. | Graphical user interface, application  Description automatically generated  This displays a message that the door is open of the fridge. |
| Storing food items and registered in a database | The system must store all food items and registered users in separate databases. | Graphical user interface, application  Description automatically generated  Shape, square  Description automatically generatedThis is where the food quantity is stored and the items of the food in the fridge is displayed. |
| System database for food | The system database for food items must have a capacity limit of 440 different types of food items (rows) so that the food items to not exceed the fridge capacity.  The system database for food items must contain the attributes, name, quantity, and expiry dates | Diagram  Description automatically generated  This shows the system database of the food which has all the details of each food item. |
| Restocking food items on systems | The system must have a manually adjustable restocking threshold for food items, | Graphical user interface, application  Description automatically generated  Graphical user interface, application  Description automatically generated  This where the system restock food items and can determine the quantity of each food item. |
| Notifying head chef when a food item falls | The system must prompt the “head chef” users computer device when a food item falls below the restocking threshold. | Graphical user interface, application  Description automatically generated  This is where the head chef gets communicated that a food item is fallen. |
| Admin user of system | The system must have an account/user that has the admin with the authority to change user authority. | Graphical user interface  Description automatically generated with medium confidence  This is the page where a system admin logs into the application. |
| System setting specific order every Monday | The system will set an order to the appropriate supplier every Monday for food items that are low or are zero in quantity. | Shape  Description automatically generatedThis demonstrates the system having a complete order on Mondays. |
| Delivery person will insert number of items of the fridge and is recorded by system | The system will record the number of items inserted into the fridge from the user “delivery person” and will compare that number to the checksum that is given before the insertion of food items | Chart, waterfall chart  Description automatically generated  Graphical user interface, application  Description automatically generated  The delivery person successfully logs in and will record the number of items in the fridge and also how they make a delivery. |
| Sending food data to health a safety officers. | Report of all the food inside the fridge will be sent to the health and safety officers. | Shape, square  Description automatically generated  The safety officers will obtain the report of the food inside the fridge |
| System records number of items that’s in fridge | The system will send a report or list of items that have been inserted into the fridge to the user that performed the inserts. | Shape, square  Description automatically generated  This is where there is detailed descriptions of food items in the fridge and shows what date they were added. |
| Understandable UI representation for the users | The system should have a UI that is easy to use and understand for all user types. | \*all pictures above\*  The UI is easy to use and is knowledgeable to all user types. |
| Hash function on the logins | System implements a hash function to user username and passwords to store and secure the information. | Table  Description automatically generated  In the manage users section of the system admin menu is where the usernames and passwords of users are secured. |
| Messaged to head chef to alert them the food has 3 days to expire. | The system prompts the “head chef” users mobile app when food items are 3 days before its expiry date. | Shape, square  Description automatically generatedThis demonstrates a message being sent to the head chef which displays what food is going to be expired. |
| System is available from a website. | System be accessible from a website | \*not included\* |

# References

Rafael D. Hernandez (2021) The Model View Controller Pattern Explained. *Available at:* <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/> *(Accessed: 08/02/2022)*

Tom Dalling (2009) Model View Controller Explained. *Available at:*

<https://www.tomdalling.com/blog/software-design/model-view-controller-explained/> *(Accessed: 08/02/2022)*

Phillipe Desfray, Gilbert Raymond (2014) Modelling Enterprise Architecture with TOGAF. *Doi:* <https://doi.org/10.1016/B978-0-12-419984-2.00008-2>

Lucid Software Inc (2021) UML sequence diagrams. *Available at:* <https://www.lucidchart.com/pages/uml-sequence-diagram> *(Accessed: 11/02/2022)*

IBM (2022) Program Structure Design. *Available at:* <https://www.ibm.com/docs/en/raa/6.1?topic=diagrams-program-structure-diagram> *(Accessed: 11/02/2022)*

# Appendix

## Appendix 1

Diagram, schematic

Description automatically generated