**Systematic and Technologic Automated Restaurant System**

**(STARS)**

**cs3337 Software Engineering**

**Software Requirements, Software Design and Software Test Plan Document**

Prepared by

Fong, William

Mahbub, Saba

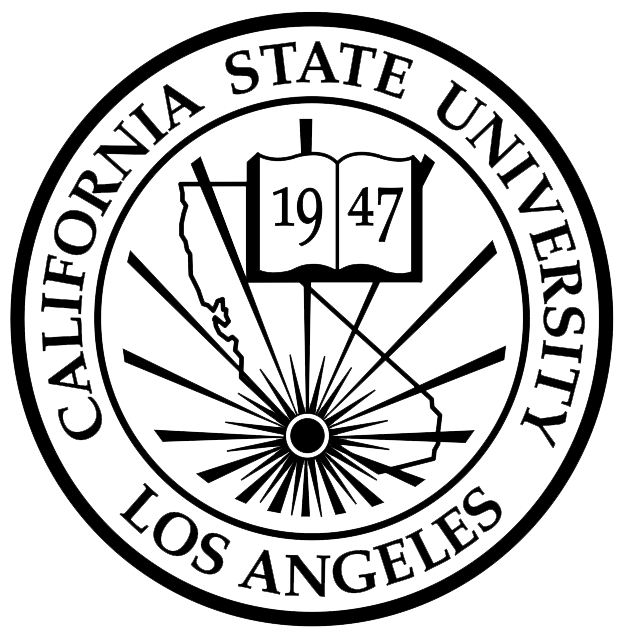
Mahbub, Safa

Placencia, Juan

November 14, 2016

CALIFORNIA STATE UNIVERSITY

LOS ANGELES



Los Angeles, California

**Systematic and Technologic Automated Restaurant System**

**(STARS)**

**Prepared By:**

Fong, William

Mahbub, Saba

Mahbub, Safa

Placencia, Juan

**Approved by:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Jose Macias Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Richard Cross Date

**Document Change Log**

|  |  |  |
| --- | --- | --- |
| **Update** | **Date Released** | **Changes** |
| Draft #1 | 10/20/2016 | Delivery of the Software Requirements document |
| Final Document | 11/14/2016 | Last update and upload to CSNS |

# List of TBD Items

|  |  |  |  |
| --- | --- | --- | --- |
| Page | **Item** | **Description** | **Status** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

**1.0 INTRODUCTION 1-3**

1.1 Purpose 1

1.2 Scope 1

1.2.1 Document Organization 1

1.2.2 Relationship to Other Documents 1-2

1.3 STARS System Architecture 2

1.3.1 Detailed Context Diagram (Level 0 DFD) 2

1.3.2 Description of STARSMajor Units 2

1.4 Requirement Development Process 2-3

1.5 References 3

1.5.1 Controlling Documents 3

1.5.2 Applicable Documents 3

1.5.3 Standards 3

**2.0 DETAILED FUNCTIONAL DESCRIPTION OF *STARS* 4-5**

2.1 Detailed STARS Functional Description 4

2.1.1 Higher Level DFDs 4

2.1.2 Detailed Functional Description of Major Sub-Units 4-5

**3.0 STARS REQUIREMENTS 6-9**

3.1 STARS FUNCTIONAL REQUIREMENTS 6-9

3.2 STARS NON-FUNCTIONAL REQUIREMENTS 9

3.3 STARS HARDWARE REQUIREMENTS 9

**4.0 STARS DETAILED DESIGN 10-15**

**5.0 STARS ELEMENTS OF IMPLEMENTATION 16-22**

**6.0 STARS TEST PLAN 23-27**

6.1 Introduction 23

6.2. Functional Requirements Validation Matrix 23-27

**A. ACRONYMS 28-29**

**B. DATA DICTIONARY 29**

**List of Figures**

Figure 1 Data Flow Diagram-Level 0 2

Figure 2 Data Flow Diagram-Level 1 4

Figure 4.1.1 Level 2 Data Flow Diagram-Main Module 10

Figure 4.1.2 Level 2 Data Flow Diagram-Menu Control Module 11

Figure 4.1.3 Level 2 Data Flow Diagram-Availability Module 12

Figure 4.1.4 Level 2 Data Flow Diagram-Authoritative Action Module 13

Figure 4.1.5 Level 2 Data Flow Diagram-Payment Module 14

Figure 4.1.6 Level 2 Data Flow Diagram-Kitchen Module 15

Figure 5.1.1 Main Module Code Sample One 16

Figure 5.1.1 Main Module Code Sample Two 17

Figure 5.2.1 Menu Control Module Code Sample 18

Figure 5.3.1 Availability Module Code Sample One 18

Figure 5.3.2 Availability Module Code Sample Two 19

Figure 5.5.1 Payment Module Code Sample One 20

Figure 5.5.2 Payment Module Code Sample Two 21

Figure 5.6.1 Kitchen Module Code Sample 22

**List of Tables**

None

**1. INTRODUCTION**

**1.1 Purpose**

The purpose of this document is **four-fold**:

a) Completely define a full set of requirements for the **STARS – Section 3.0**.

b) Completely define the design for the **STARS – Section 4.0**.

c) Define and partially implement feasible modules for the **STARS – Section 5.0**.

d) Completely define the Test Plan for the **STARS – Section 6.0**.

The complete definition of all **STARS** requirements provides the source requirement inputs for the development of the subsequent supporting software subsystems documents.

**1.2 Scope**

The documentation developed as part of this cs3337 class, starts with the SRD including elements of Software Design and parts of a Test Plan.

The scope of this document includes the following:

* All functional and nonfunctional requirements on the **STARS** are captured. This includes Verification & Validation (V&V) requirements, as well as inter-software subsystems requirements.
* A complete set of **STARS** Requirements. These requirements are organized by key **STARS** functional units shown on the Level 1 DFD. The Level 1 DFD is shown on

page 4.

* A trace matrix, relating all **STARS** functional requirements to functional subunits as expanded in lower level DFDs. Level 2 and higher DFDs are provided on pages 10-15.
* The functional requirements defined in the **STARS** Requirements section have been expanded to include more specific hardware requirements.

**1.2.1 Document Organization**

The organization of this document provides a natural flow or allocation of requirements to each succeeding section.

Details regarding the overall document are given in sub-section 1.5 below.

**1.2.2 Relationship to Other Documents**

The **STARS** SRD/SDD/STP/SID is a complete self-contained document. Some relationships to other documents in the literature are indicated below in subsection 1.5.

**1.3 STARS Architecture**

1.3.1 Detailed Context Diagram (DFD Level 0)

The **STARS** architecture is summarized in the Context Diagram (DFD Level 0) given below. A more complete Functional Description is given in Section 2 of this document. The Context Diagram provides the overall structure of the software modules and all its inputs and outputs. The notation used corresponds to that defined for any Data Flow Diagram (DFD).

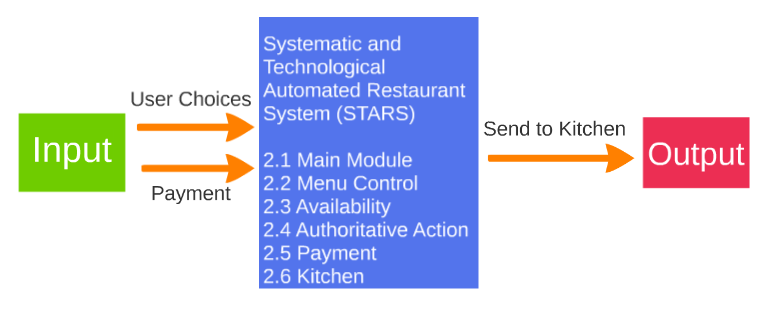


Figure 1: Data Flow-Diagram Level 0

**1.3.2** Description and major functions of **STARS**

STARS automate the technical aspects of restaurants that shall be self-sufficient. The system is provided to the user (the customer in the restaurant) so that they may choose what food or drink they want, pay for it, and wait to be served without contact with anyone else, through a touch screen interface. The interface is connected to databases of available food options to show real time availability. STARS make it possible for a restaurant experience with as little interaction with other humans but there will be options for the user to signal for help from the manager or other on-hand staff.

**1.4 Requirement Development Process**

The **STARS** detailed functional description is documented in section 2.0. Basically, Section 2 is a succinct software description document. The overall detailed functional description is based on higher level DFDs (above level 1). All major functional units are described in detail in this part of the document.

In general, all requirements affecting **STARS** are captured in Section 3.0 of this document. These requirements are a refinement and completion of requirements first collected as part of this Software Engineering project. The document is cited in Section 1.2.2. This section is the one worked in most detail to become a reasonably complete Software Requirements Document (SRD). It includes both functional and non-functional software requirements together with several detailed “rational” paragraphs whenever necessary to complete the understanding of each requirement.

Section 4 is the detailed **STARS** Software Design Description Document (SDD). This part of the document includes all higher-level DFDs as described in section 2 plus all interface units. The document is highly technical and it is based on section 2 descriptions. An important component is the addition of a SIS (software interface specification) document in subsection 4.2.

Section 5 includes elements of a partial implementation of **STARS**. This section includes the various constraints that effectively limit the implementation as well as the sub-units that will be coded. The implementation goals are defined and the code and pseudo code are included as an attachment to this section.

Section 6 is the last major section in this document and includes the overall Test Plan (TP) of the **STARS**. The test plan details the various techniques used to test the requirements and it also includes a Validation Matrix where each requirement specified in section 3 is listed with its corresponding validation method. The validation methods may include Testing, Analysis and Demonstration, and possible other V&V methods. In addition, the TP specifies the mandated peer reviews needed to validate the stakeholder’s part of the requirements.

**1.5 References**

All references used in the creation of this document are listed below.

**1.5.1 Controlling Documents**

1) There is no document controlling this document.

**1.5.2 Applicable Documents**

1) The template document provided through email, in class, was used in the production of this document.

**1.5.3 Standards**

No Standard has been used in the creation of this document. However, some Standards described in textbooks have been examined as a reference. In particular, the IEEE standard has been briefly discussed in class.

**2.0 DETAILED FUNCTIONAL DESCRIPTION OF *STARS***

2.1 Detailed **STARS** Functional Description.

The major tool used to design **STARS** is the Data Flow Diagram, DFD. The rationale behind the selection of DFDs as the preferred design tool was their simplicity and versatility. In the future more sophisticated tools may be used particularly if a correlation from Design to Requirement to Implementation and Testing is found to be a necessary addition.

2.1.1 Higher Level Data Flow Diagrams.

The **STARS** major functional design components are shown in the DFDs below.

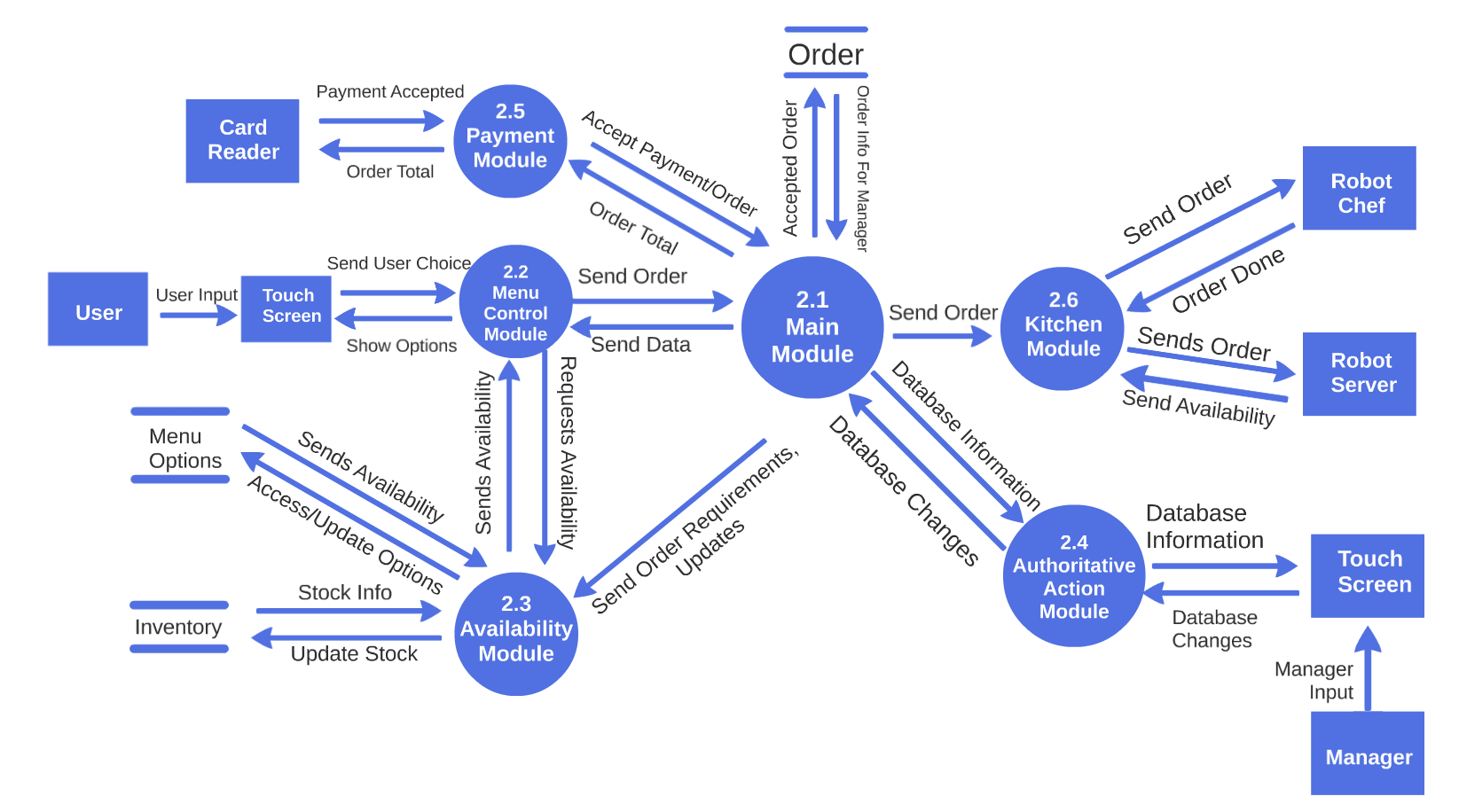


Figure 2: Data Flow Diagram-Level 1

2.1.2 Detailed Functional Description of **STARS** Major Sub-Units

The **STARS** major functional subunits shown in the DFDs in the previous sub-section are described in detail below.

**Main Module (MM) - Module 2.1**

The Main Module (MM) is the central processing module that links the other modules together to keep information flow smooth and efficient. The MM is charged with taking requests and gives back the appropriate response. The MM will also take transaction conformations and send them to a database and kitchen module. The MM contains sub modules to help it process all the data.

**Menu Control Module (MCM) - Module 2.2**

The Menu Control Module (MCM) connects the user to the program through a touch screen. The MCM will receive input from the user and output information to the user through the touch screen. It will also send the order to the MM once the user finalizes the order. The MM will send various data to the MCM such as any status messages, errors, etc.

**Availability Module (AM)- Module 2.3**

The Availability Module (AM) will provide and update information about the current inventory of the ingredients of the menu items. Every time an order is successfully placed, the AM will receive data from the main module to update the new inventory stock.

The AM will also read the availability from the inventory and update which items on the menu are available to be ordered and send it to the MCM to be displayed.

**Authoritative Action Module (AAM) - Module 2.4**

The Authoritative Action Module (AAM) is used when a manager needs to read information/change information within the three databases.

**Payment Module (PM)- Module 2.5**

The Payment Module (PM) handles the all the payment operations for the restaurant. Once the user has selected that they are done ordering from the User GUI/UI, the total order shall be sent to be processed by the module.

**Kitchen Module (KM)- Module 2.6**

The Kitchen Control Module (KM) handles the cooking and serving aspects of the automated restaurant. The module will take approved orders (orders with confirmed payments) and will then distribute the information to the external Robot Chefs (RC) and external Robot Server (RS). It is noted that the RC and RS are both external devices and as such does not interact with the software much. The only data retrieved by the KM is when the RC has finished cooking their order and which RS are available to serve.

**3.0 *STARS* REQUIREMENTS**

3.1 STARS Functional Requirements

This Section collects all **STARS** Functional Requirements. The Section includes the complete set of functional requirements with explanation and rational where the statement of the requirement was deemed insufficient or needing additional background/justification. All requirements relate to the design modules described in Section 2. An effort has been made to standardize the correlation between the design modules and the requirements to make their access and organization more consistent. For example, module 2.1 requirements are labeled 3.1; sub-module 2.1.1 requirements are labeled 3.1.1 and so on. The list of requirements follows.

|  |  |
| --- | --- |
| Requirements Related to Design Module 2.1 Main Module (MM) | |
| Requirement No. | Requirement Description |
| 3.1-1 | MM shall consist of two: Menu Options and Order List. |
| 3.1-2 | MM shall provide a tabbed window format for each element. |
| 3.1-3 | MM shall provide the necessary controls for all aspects of the hardware. |
| 3.1-4 | MM shall display any necessary error, or other messages to the user. |
| 3.1-5 | MM shall display all possible menu items to the user. |
| 3.1-6 | MM shall display all menu items listed in the orders tab to the user. |
| 3.1-7 | MM shall display a comments section in the orders tab to the user. |

|  |  |
| --- | --- |
| Requirements Related to Design Module 2.2 Menu Control Module (MCM) | |
| Requirement No. | Requirement Description |
| 3.2-1 | MCM shall process all menu options |
| 3.2-2 | MCM shall send available menu options to the MC |
| 3.2-3 | MCM shall send the user's Orders to the MC. |
| 3.2-4 | MCM shall process the ingredients used. |
| 3.2-5 | MCM shall update the Menu Options database. |

|  |  |
| --- | --- |
| Requirements Related to Design Module 2.3 Availability Module (AM) | |
| Requirement No. | Requirement Description |
| 3.3-1 | AM shall process orders from the MC. |
| 3.3-2 | AM shall update the inventory in the inventory database. |
| 3.3-3 | AM shall process the Inventory Database. |
| 3.3-4 | AM shall update the Menu Options database from the information collected by the Inventory Database. |
| 3.3-5 | AM shall inform the MC of the changes to the Database. |

|  |  |
| --- | --- |
| Requirements Related to Design Module 2.4 Authoritative Action Module (AAM) | |
| Requirement No. | Requirement Description |
| 3.4-1.1 | AAM shall process Database changes sent by the manager. |
| 3.4-1.2 | AAM shall give permission to the Manager to handle the Changes. |
| 3.4-1.3 | GUI Display Module (GDM) |
| 3.4-1.4 | GDM shall display to the manager the changes that they are able to do. |
| 3.4-1.5 | GDM shall process the changes the Manager inputs. |
| 3.4-1.6 | GDM shall send changes to the appropriate Modules for the database changes to occur. |
| 3.4-1.7 | GDM shall process information from modules. |
| 3.4-1.8 | GDM shall display the changes to the Manager. |
| 3.4-1.9 | Orders Update Module (OUM) |
| 3.4-2.0 | OUM shall process information given to it from the GDM. |
| 3.4-2.1 | OUM shall send the changes to Databases to make the changes. |
| 3.4-2.2 | Inventory Update Module (IUM) |
| 3.4-2.3 | IUM shall process information given to it from the GDM. |
| 3.4-2.4 | IUM shall send the changes to Databases to make the changes. |
| 3.4-2.5 | Menu Update Module (MUM) |
| 3.4-2.6 | MUM shall process information given to it from the GDM. |
| 3.4-2.7 | MUM shall send the changes to Databases to make the changes. |
| 3.4-2.8 | Menu Update Module (MUM) |
| 3.4-2.9 | MUM shall process information given to it from the GDM. |
| 3.4-3.0 | MUM shall send the changes to Databases to make the changes. |

|  |  |
| --- | --- |
| Requirements Related to Design Module 2.5 Payment Module (PM) | |
| Requirement No. | Requirement Description |
| 3.5-1 | PM shall inform the Card Reader the total amount of the order. |
| 3.5-2 | PM shall handle confirmation of payment. |
| 3.5-3 | PM shall handle declining of payment. |
| 3.5-4 | PM shall tell the MC if the order has been paid for. |

|  |  |
| --- | --- |
| Requirements Related to Design Module 2.6 Kitchen Module (KM) | |
| Requirement No. | Requirement Description |
| 3.6-1 | KM shall process all orders sent to it. |
| 3.6-2 | KM shall send the order to the Robot Chef to produce the food. |
| 3.6-3 | KM shall receive a menu update from the Robot Chef. |
| 3.6-4 | KM shall send MC the progress of the order. |

**3.2 *STARS* Non-Functional Requirements**

This Section collects all the Project-Acronym Non-Functional Requirements. All non-functional requirements are numbered “NF – n” where “n” indicates the nth requirement.

NF - 1 STARS shall display information in real time.

**3.3 *STARS* Hardware Requirements**

This Section collects all the Project-Acronym Hardware Requirements. All hardware requirements are numbered “H – n” where “n” indicates the nth requirement.

H - 1 STARS will run on standard microcomputer hardware.

H - 2 STARS will require a display device. Each table will have a touch screen display for user input and screen capability, suitable for text and image displays.

H - 3 STARS will connect to a card reader that is attached to the touch screen. Any type of debit/credit card shall be able to be read by the card reader including but not limited to Visa, MasterCard, American Express, etc. (should we include apple pay and other means of payment).

H - 4 STARS will be connected to a high tech, futuristic Robot that is able to read data and process it to make food. Robot shall be able to sense/scan where ingredient are located and through visual scanning measure the correct amounts needed.

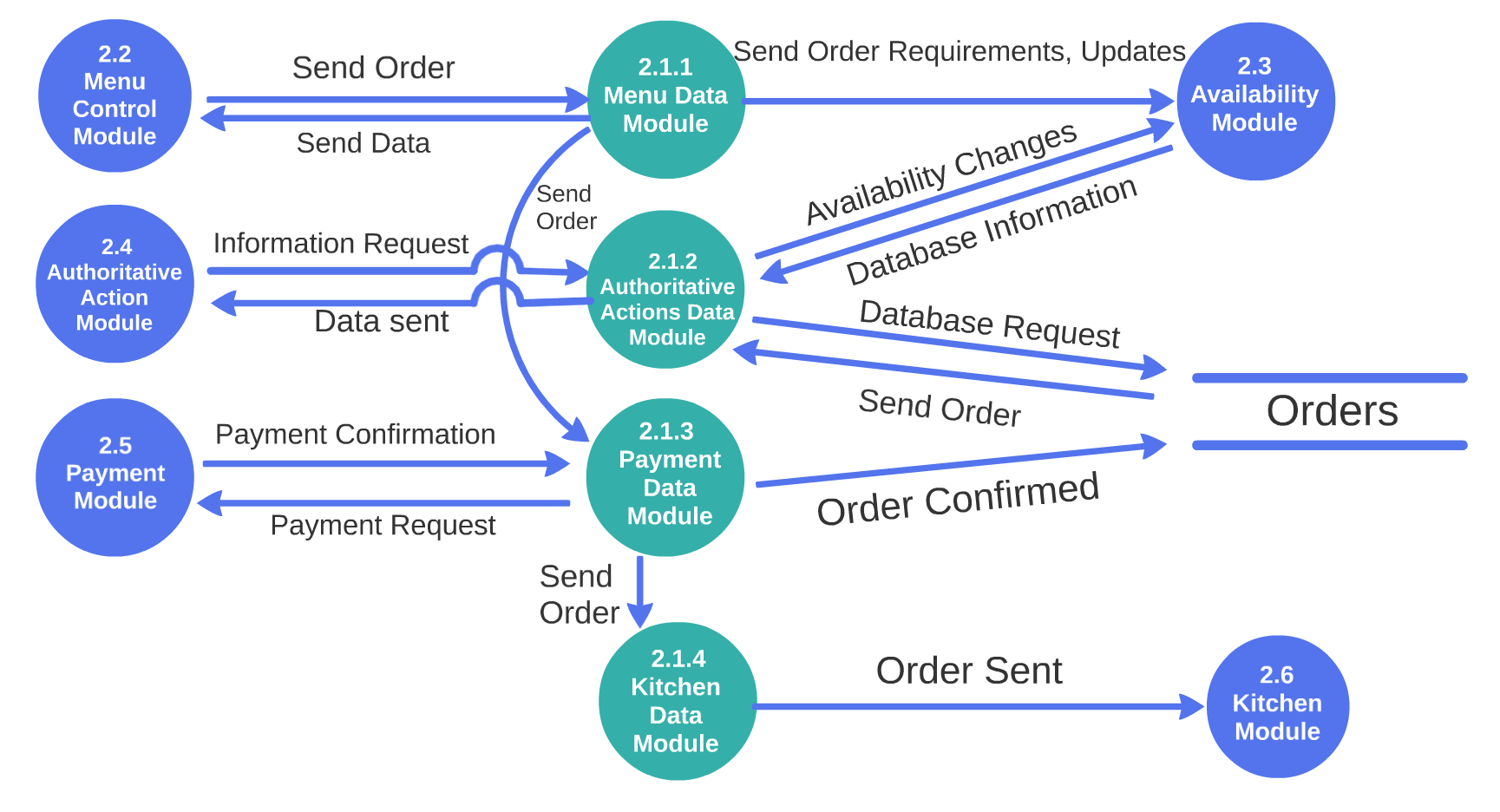
H - 5 STARS will be connected to a high tech, futuristic Robot Waiter (RW) which will have scanners that allow it to sense people, table, and plates. Once the RW is handed food, it will walk over to the table indicated.

H - 6 STARS will require a display device. There will be a separate touch display for manager input and screen capability, suitable for text and image displays.

**4.0 *STARS* DETAILED DESIGN**

In this section the **STARS** described in Section 2 with requirements listed in Section 3 will be designed in detail possibly including higher level DFDs. Each major module detailed design is included in correspondence with the design sections defined in Section 2 and responding to the requirements listed in its correlated sub-section in chapter 3.

**Level 2 DFD 2.1 Main Module**

Figure 4.1.1: Level 2 Data Flow Diagram-Main Module

2.1.1 Menu Data Module

The Menu Data Module (MDM) is responsible for sending data from the Main Control Module to other modules to keep data flow consistent. The MDM will receive an order and send it to the Availability Module to check if the order can be made. Once the order is confirmed, it gets sent to the Payment Data Module.

2.1.2 Authoritative Actions Data Module

The Authoritative Actions Data Module (AADM) is in charge of giving the Authoritative Actions Module database information so the user can make the necessary changes or keep track on what’s going on. The AADM will get a request from the Authoritative Actions Module, and then it acquires that information for the Authoritative Actions Module. If ordered, the AADM can make changes in the Availability Module.

2.1.3 Payment Data Module

The Payment Data Module (PDM) makes sure the confirmed orders are paid for. The PDM gets an order from the MDM then sends it to the PM to get paid for. Once payment is confirmed, the order gets stored in the orders database and gets send to the Kitchen Data Module.

2.1.4 Kitchen Data Module

The Kitchen Data Module (KDM) takes the confirmed order from the PDM and sends it to the Kitchen Module.

**Level 2 DFD 2.2 Menu Module**

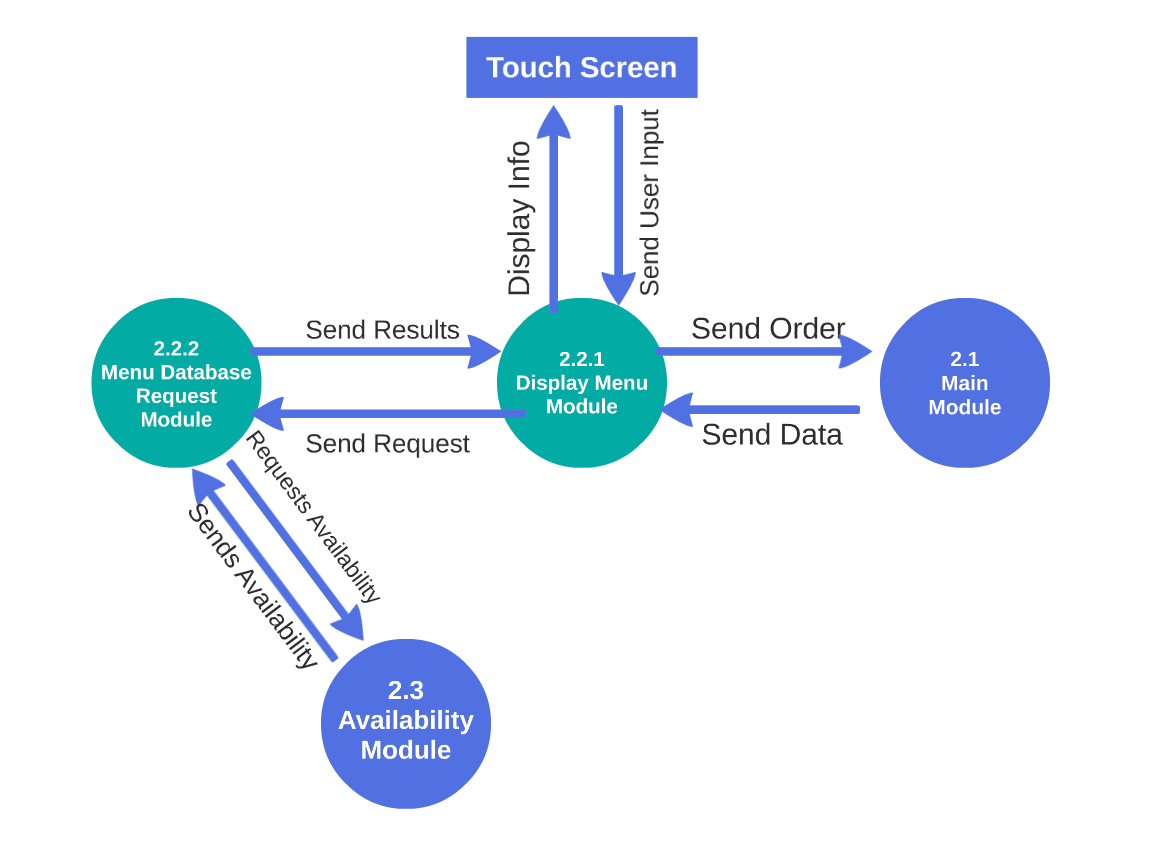


Figure 4.1.2: Level 2 Data Flow Diagram-Menu Control Module

2.2.1 Display Menu Module

The Display Menu Module (DMM) is used to interact with the user and relay information between the system and the user.

The DMM takes input from the user through the touch screen and also displays the necessary information on the touch screen for the user. The DMM sends a request to the Menu Database Request Module whenever the user is ordering to get the menu items that are available.

After it takes the order from the user, it sends the completed order to the MM.

2.2.2 Menu Database Request Module

The Menu Database Request Module (MDRM) is used to request for information regarding the availability of items on the menu.

The MDRM takes a request from the DMM to request information from the Availability Module. It sends the request and receives back the results of the available items to send back to the DMM.

**Level 2 DFD 2.3 Availability Module**

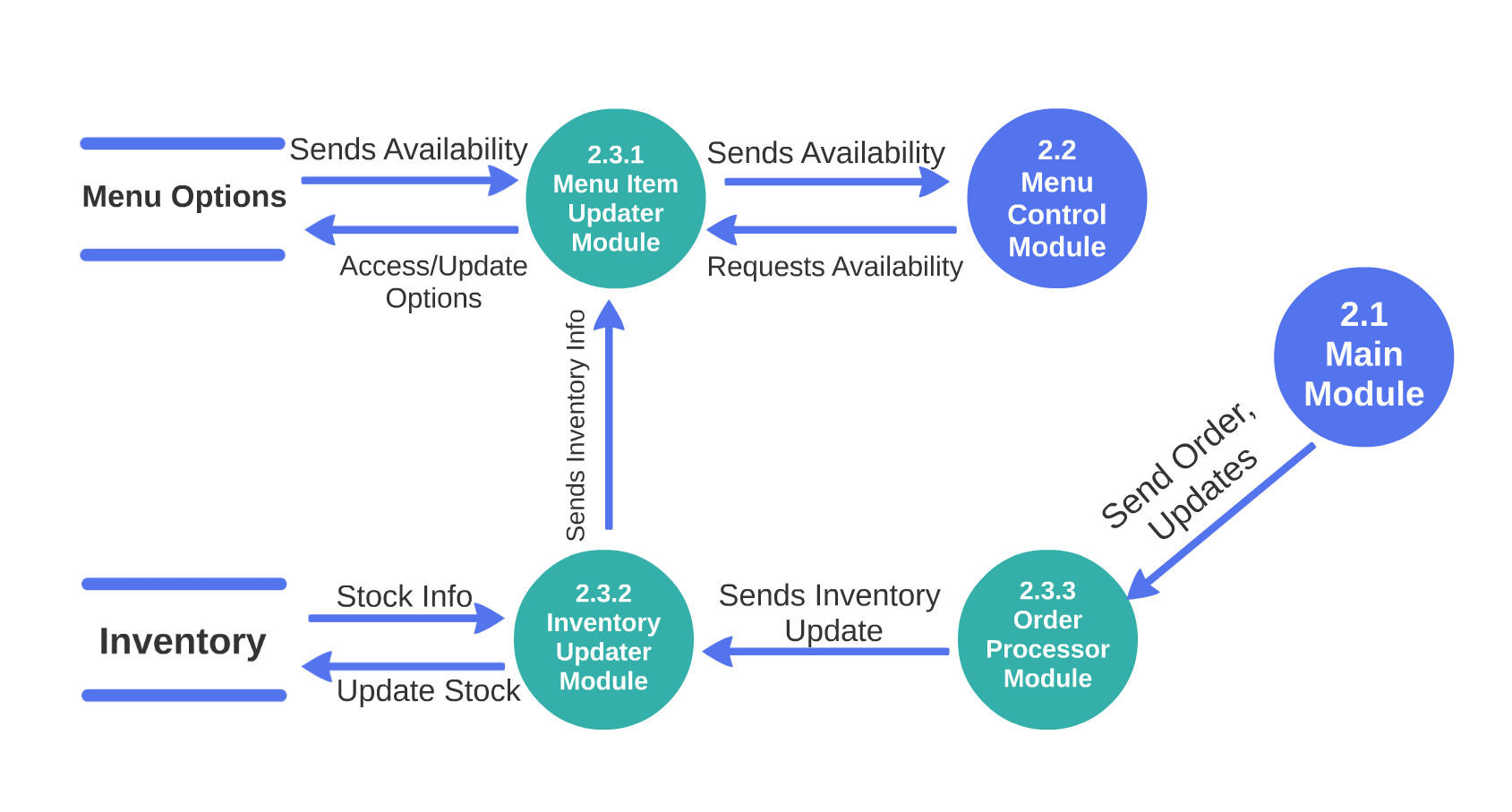


Figure 4.1.3: Level 2 Data Flow Diagram-Availability Module

2.3.1 Menu Item Updater Module

The Menu Item Updater Module (MIUM) is used to process information about the availability of the menu items.

The MIUM receives requests from the MCM, which will then check the Menu Options database on the availability of the menu items. The MIUM then sends the results back to the MCM.

The MIUM also receives information about the inventory from the Inventory Updater Module. The MIUM will take this data and update the Menu Options database accordingly.

2.3.2 Inventory Updater Module

The Inventory Updater Module (IUM) is used to process information about the availability of the inventory.

The IUM takes information from the Order Processor Module about inventory updates and updates the Inventory database with this information. The IUM also sends the inventory information to the MIUM whenever there are any changes that are made.

2.3.3 Order Processor Module

The Order Processor Module (OPM) processes information about completed orders and any updates needed to be made.

The OPM takes in the order or updates from the MM. From that information, the OPM dissects the ingredients that were used in the order, and it sends that information to the IUM to be updated in the database.

**Level 2 DFD 2.4 Authoritative Action**

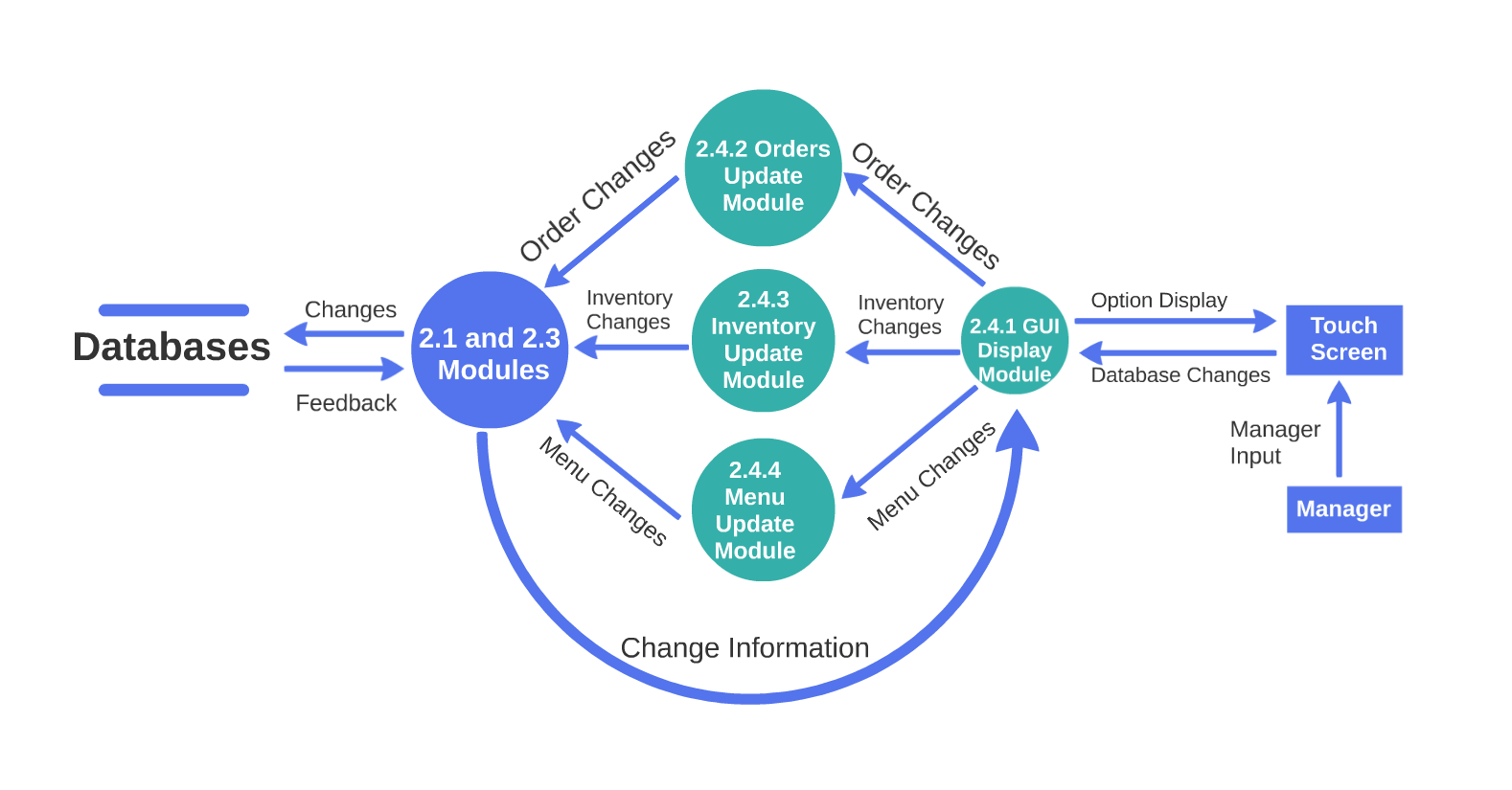


Figure 4.1.4: Level 2 Data Flow Diagram-Authoritative Action Module

2.4.1 GUI Display Module

The GUI Display Module (GDM) validates that the managers is authorized to make changes to the database. Then the GUI will give the manager the option to change whichever database the manager wants to change. Depending on the Database chosen, different changes will be available.

2.4.2 Orders Update Module (OUM)

The Orders Update Module (OUM) deals with changes in the Orders database. The AAM sends the orders to the Manager UI. The manager is then able to sort through the information provided to deal with financial needs or customer request. The manager is only able to append information to the order, not directly change the names, amount, confirmation, or complaints.

2.4.3 Inventory Change Module (ICM)

The Inventory Change Module (ICM) deals with changes in Inventory database. The AAM sends the items in the inventory to the Manager UI. The manager is then able to sort through the items and put in value changes. The manager is able to add items, delete items, and change quantity in this Module.

2.4.4 Menu Update Module (MUM)

The Menu Update Module (MUM) deals with changes in the Menu Options database. The AAM sends the menu options to the Manager UI. The manager is then able to add menu items. The Manager will have to input the ingredients needs, directions, and recipe as well for the Robot Chef to process the information.

**Level 2 DFD 2.5 Payment Module**

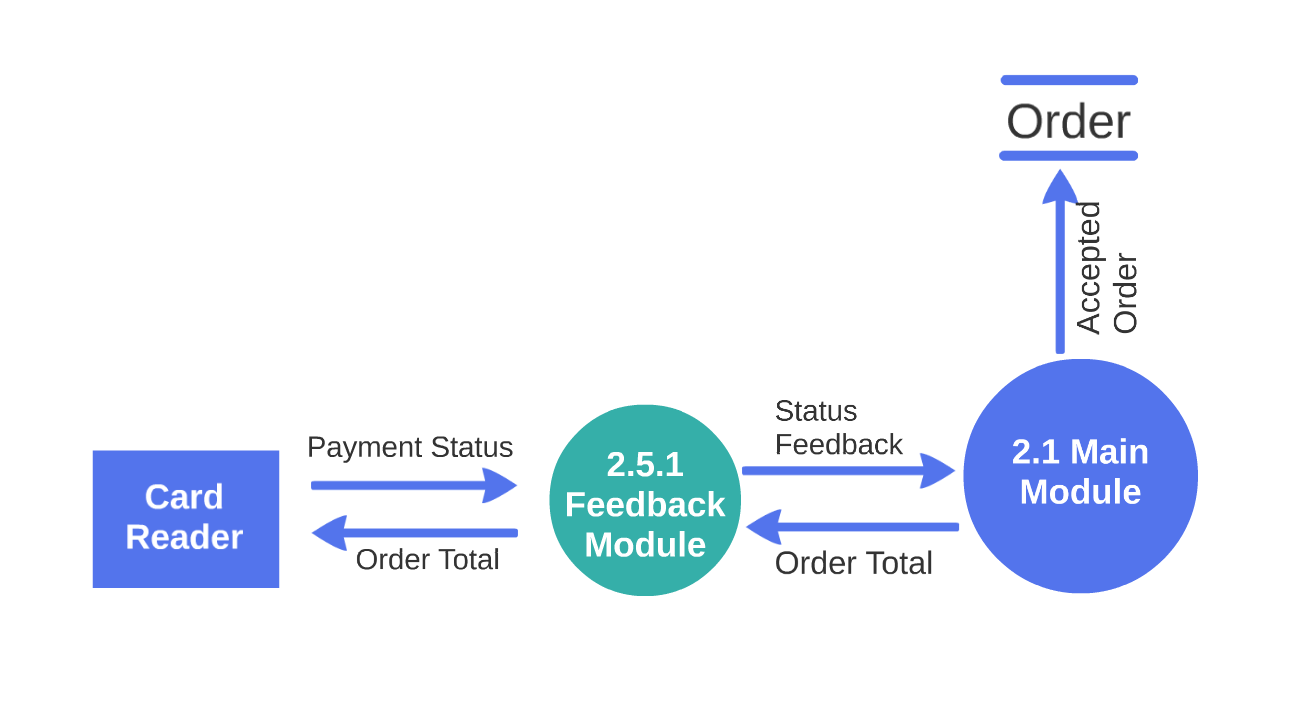


Figure 4.1.5: Level 2 Data Flow Diagram-Payment Module

2.5.1 Feedback Module (FM)

The FM shall validate that the Orders are okay, and then shall send the payment amount to an external card reader. The external card reader is not part of our program. The only data PM retrieves from the card reader is Payment approval or decline. Should the payment be declined a message shall be sent to the User through the GUI/UI indicating the error. Should the payment be approved the PM shall send this information to the Main Control (MC) which will use this information to process other operations.

**Level 2 DFD 2.6 Kitchen Module**

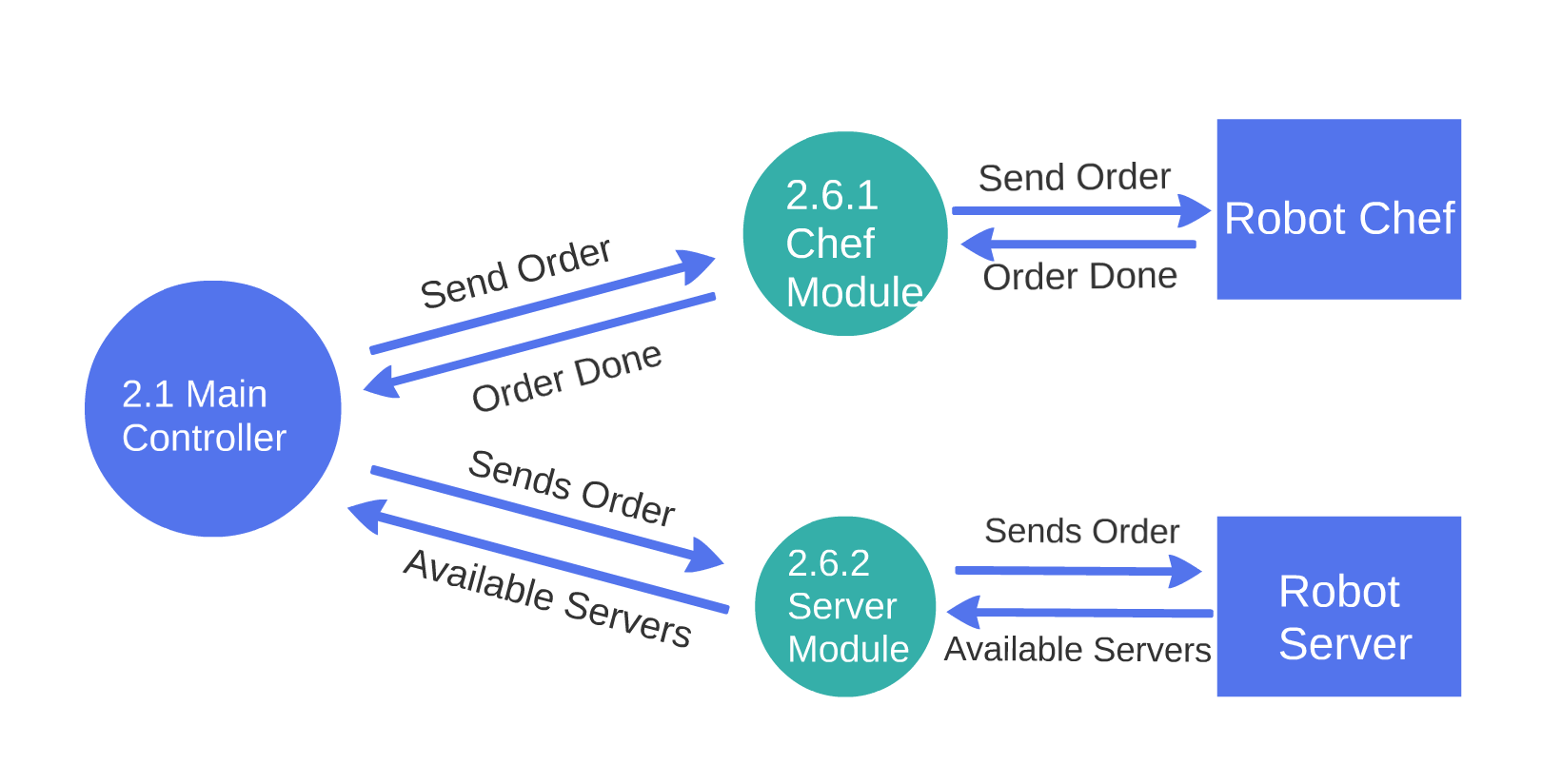


Figure 4.1.6: Level 2 Data Flow Diagram-Kitchen Module

2.6.1 Chef Module

The MM shall send the order to the Chef Module (CM) that shall send the order to the RC. Once the food is cooked properly, the RC shall send the information that the food is ready to the MM.

2.6.2 Server Module

When MM receives information from CM that the food is cooked, MM shall tell the Server Module(SM). SM shall tell the RS to take the order and serve it to the customer. RS shall return to its waiting station and tell RS that it is available to serve again.

**5.0 *STARS* ELEMENTS OF IMPLEMENTATION**

In this section (some of) the modules designed in Section 4 with requirements listed in Section 3 will be implemented initially at least at the level of pseudo code. Where possible, actual code will be provided. The entirety of the code produced is found on the website “GitHub,” the link is: <https://github.com/WilliamFong/STARS>. For clarity, each module is implemented in correspondence with the design sections defined in chapter 2 and responding to the requirements listed in its correlated sub-section in chapter 3. The following elements of implementation will have figures displaying code that is the foundation of the corresponding module.

5.1 STARS Module 2.1: Main Module

This module was not implemented completely. The main module deals with data transfer between all modules and all modules were not implemented, thus the Main Module was not implemented completely. The implementation of the current project is given below.

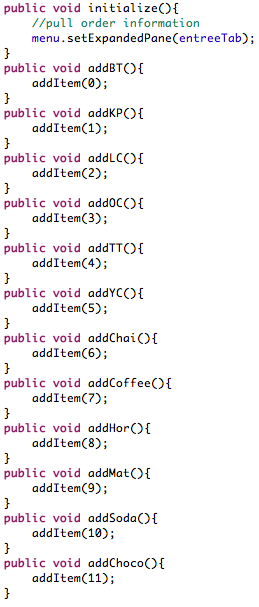


Figure 5.1.1: Main Module Code Sample One



Figure 5.1.1: Main Module Code Sample Two

5.2 STARS Module 2.2: Menu Control Module

This module was not implemented completely. The Menu Control Module determines what the user sees on the touch screen when they are ordering their food and is meant to be dynamic according to whether the menu items are available. The Availability Module determines what menu options are available but was not implemented. Thus the Menu Control Module is not dynamic and not implemented completely. The implementation of the current project is given below.

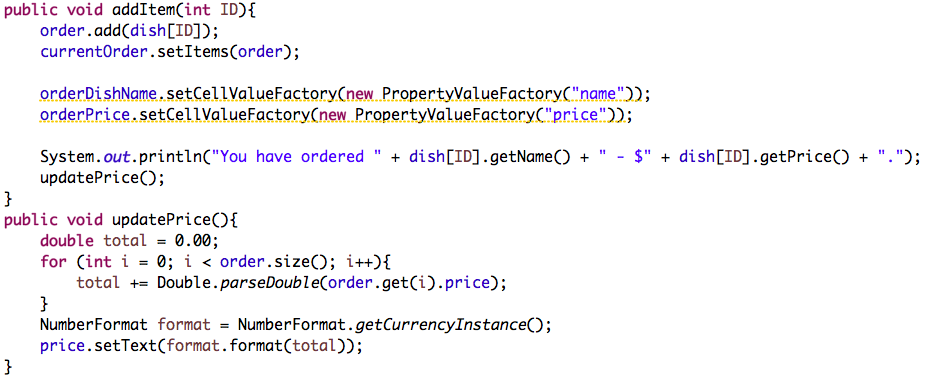


Figure 5.2.1: Menu Control Module Code Sample

5.3 STARS Module 2.3: Availability Module

This module was not implemented. However to accommodate future updates to the software, an availability option is integrated into the basic java classes used to order food items.

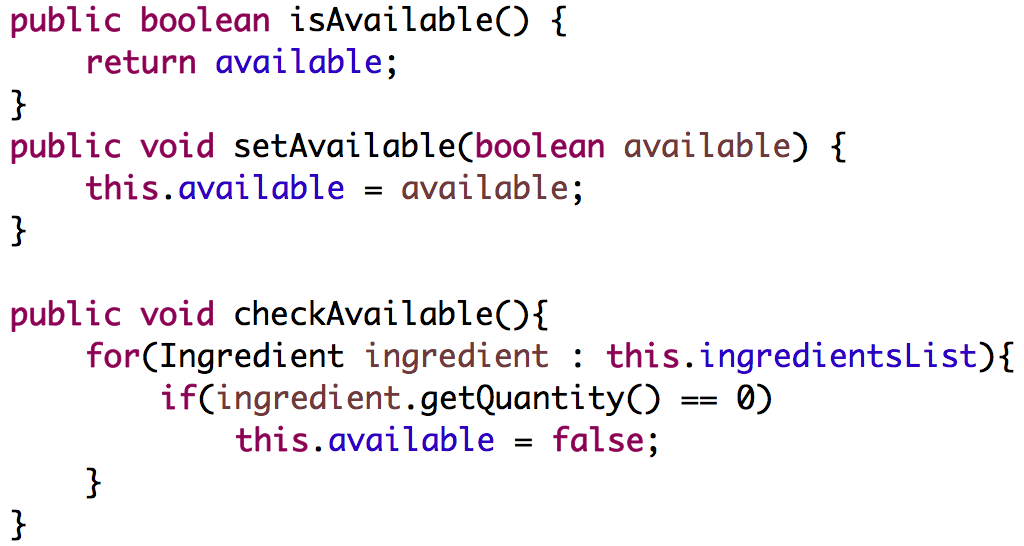


Figure 5.3.1: Availability Module Code Sample One

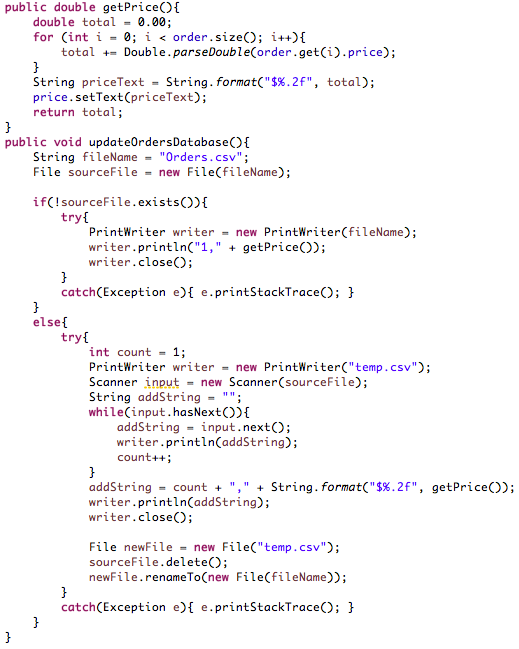


Figure 5.3.2: Availability Module Code Sample Two

5.4 STARS Module 2.4: Authoritative Action Module

This module was not implemented.

5.5 STARS Module 2.5: Payment Module

This module was not fully implemented. There is no hardware we could use to manually count and accept cash, read card information, and accept payment through phone. Our program is set up so that once this hardware is available, the customer simply has to choose which method of payment they want and as such the payment will be read.

We were not able to implement a payment error message incase there was no valid payment.

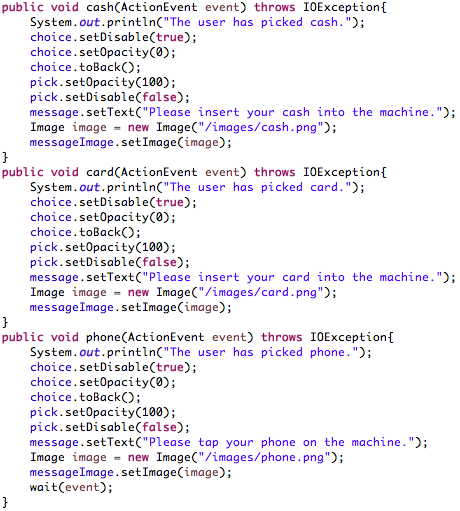


Figure 5.5.1: Payment Module Code Sample One

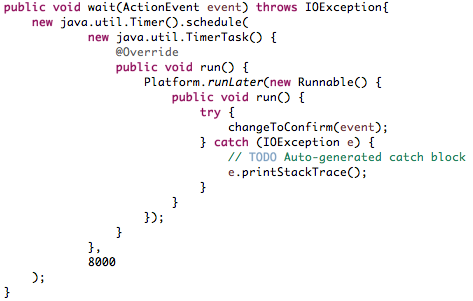


Figure 5.5.2: Payment Module Code Sample Two

5.6 STARS Module 2.6: Kitchen Module

This module was not fully implemented because we do not have the futuristic technology that would make robot chefs and robot servers available. However we set up a confirmation page as a place holder for this module, so when this screen occurs the order of the food will be sent to the robots.

**** Figure 5.6.1: Kitchen Module Code Sample

**6.1 STARS TEST PLAN**

6.1 INTRODUCTION

In this section the testing methodology to be used to V&V each of the requirements listed in section 3.0 has been identified. At points some additional testing may be required and they shall be documented as an attachment to this document.

The methodologies and testing strategies identified at this point include four major approaches: TESTING, DEMONSTRATION, INSPECTION, and ANALYSIS with various variations to adapt to the ***STARS*** characteristics:

* **Testing** using additional ad-hoc created software including a correlation-testing unit.
* **Demonstration** of the specified capability
* **Inspection** of the software code possibly using additional inspection techniques
* **Analysis** of the specific code operation/algorithm to prove functionality.

6.2 FUNCTIONAL REQUIREMENTS VALIDATION MATRIX

The *STARS* Functional and Performance Requirements Validation Matrix is given below.

|  |  |  |
| --- | --- | --- |
| V&V Related to Design Module 2.1 and Sub-modules 2.1.1, .2.1.2, 2.1.2, 2.1.3, and 2.1.4 | | |
| Requirement No. | Requirement Description | V&V Methodology |
| 3.1-1 | MM shall consist of two: Menu Options and Order List. | Not yet tested |
| 3.1-2 | MM shall provide a tabbed window format for each element. | Not yet tested |
| 3.1-3 | MM shall provide the necessary controls for all aspects of the hardware. | Not yet tested |
| 3.1-4 | MM shall display any necessary error, or other messages to the user. | Not yet tested |
| 3.1-5 | MM shall display all possible menu items to the user. | Not yet tested |
| 3.1-6 | MM shall display all menu items listed in the orders tab to the user. | Not yet tested |
| 3.1-7 | MM shall display a comments section in the orders tab to the user. | Not yet tested |

|  |  |  |
| --- | --- | --- |
| V&V Related to Design Module 2.2 and Sub-modules 2.21, and 2.2.2 | | |
| Requirement No. | Requirement Description | V&V Methodology |
| 3.2-1 | MCM shall process all menu options | Not yet tested |
| 3.2-2 | MCM shall send available menu options to the MC | Not yet tested |
| 3.2-3 | MCM shall send the user's Orders to the MC. | Not yet tested |
| 3.2-4 | MCM shall process the ingredients used. | Not yet tested |
| 3.2-5 | MCM shall update the Menu Options database. | Not yet tested |

|  |  |  |
| --- | --- | --- |
| V&V Related to Design Module 2.3 and Sub-modules 2.3.1, 2.3.2, and 2.3.3 | | |
| Requirement No. | Requirement Description | V&V Methodology |
| 3.3-1 | AM shall process orders from the MC. | Not yet tested |
| 3.3-2 | AM shall update the inventory in the inventory database. | Not yet tested |
| 3.3-3 | AM shall process the Inventory Database. | Not yet tested |
| 3.3-4 | AM shall update the Menu Options database from the information collected by the Inventory Database. | Not yet tested |
| 3.3-5 | AM shall inform the MC of the changes to the Database. | Not yet tested |

|  |  |  |
| --- | --- | --- |
| V&V Related to Design Module 2.4 and Sub-modules 2.4.1, 2.4.2, 2.4.3, and 2.4.4 | | |
| Requirement No. | Requirement Description | V&V Methodology |
| 3.4-1.1 | AAM shall process Database changes sent by the manager. | Not yet tested |
| 3.4-1.2 | AAM shall give permission to the Manager to handle the Changes. | Not yet tested |
| 3.4-1.3 | GUI Display Module (GDM) | Not yet tested |
| 3.4-1.4 | GDM shall display to the manager the changes that they are able to do. | Not yet tested |
| 3.4-1.5 | GDM shall process the changes the Manager inputs. | Not yet tested |
| 3.4-1.6 | GDM shall send changes to the appropriate Modules for the database changes to occur. | Not yet tested |
| 3.4-1.7 | GDM shall process information from modules. | Not yet tested |
| 3.4-1.8 | GDM shall display the changes to the Manager. | Not yet tested |
| 3.4-1.9 | Orders Update Module (OUM) | Not yet tested |
| 3.4-2.0 | OUM shall process information given to it from the GDM. | Not yet tested |
| 3.4-2.1 | OUM shall send the changes to Databases to make the changes. | Not yet tested |
| 3.4-2.2 | Inventory Update Module (IUM) | Not yet tested |
| 3.4-2.3 | IUM shall process information given to it from the GDM. | Not yet tested |
| 3.4-2.4 | IUM shall send the changes to Databases to make the changes. | Not yet tested |
| 3.4-2.5 | Menu Update Module (MUM) | Not yet tested |
| 3.4-2.6 | MUM shall process information given to it from the GDM. | Not yet tested |
| 3.4-2.7 | MUM shall send the changes to Databases to make the changes. | Not yet tested |
| 3.4-2.8 | Menu Update Module (MUM) | Not yet tested |
| 3.4-2.9 | MUM shall process information given to it from the GDM. | Not yet tested |
| 3.4-3.0 | MUM shall send the changes to Databases to make the changes. | Not yet tested |

|  |  |  |
| --- | --- | --- |
| V&V Related to Design Module 2.5 | | |
| Requirement No. | Requirement Description | V&V Methodology |
| 3.5-1 | PM shall inform the Card Reader the total amount of the order. | Not yet tested |
| 3.5-2 | PM shall handle confirmation of payment. | Not yet tested |
| 3.5-3 | PM shall handle declining of payment. | Not yet tested |
| 3.5-4 | PM shall tell the MC if the order has been paid for. | Not yet tested |

|  |  |  |
| --- | --- | --- |
| V&V Related to Design Module 2.6 and Sub-modules 2.6.1, and 2.6.2 | | |
| Requirement No. | Requirement Description | V&V Methodology |
| 3.6-1 | KM shall process all orders sent to it. | Not yet tested |
| 3.6-2 | KM shall send the order to the Robot Chef to produce the food. | Not yet tested |
| 3.6-3 | KM shall receive a menu update from the Robot Chef. | Not yet tested |
| 3.6-4 | KM shall send MC the progress of the order. | Not yet tested |

**A. ACRONYMS**

**AADM** Authoritative Actions Data Module

**AAM** Authoritative Action Module

**AM** Availability Module

**CM** Chef Module

**DFD** Data Flow Diagram

**DMM** Display Menu Module

**GDM** GUI Display Module

**ICM** Inventory Change Module

**IUM** Inventory Updater Module

**KDM** Kitchen Data Module

**KM** Kitchen Module

**MCM** Main Control Module

**MDM** Menu Data Module

**MDRM** Menu Data Request Module

**MIUM** Menu Item Updater Module

**MM** Main Module

**MUM** Menu Update Module

**OPM** Order Processor Module

**OUM** Orders Update Module

**PDM** Payment Data Module

**PM** Payment Module

**RC** Robot Chefs

**RS** Robot Server

**SM** Server Module

**STARS** Systematic and Technologic Automated Restaurant System

**B. DATA DICTIONARY**