Software Requirements Specification  
**Table of Contents**Table of Contents...............................................................................................................................................i  
List of Figures...................................................................................................................................................ii  
1.0. Introduction..............................................................................................................................................iii  
1.1. Purpose.................................................................................................................................................iii  
1.2. Scope of Project....................................................................................................................................iii  
1.3. Glossary................................................................................................................................................iv  
1.4. References..............................................................................................................................................v  
1.5. Overview of Document..........................................................................................................................v  
2.0. Overall Description....................................................................................................................................1  
2.1 System Environment...............................................................................................................................1  
2.2 Functional Requirements Specification..................................................................................................2  
2.2.1 Student Use Cases............................................................................................................................3  
2.3 User Characteristics................................................................................................................................8  
2.4 Non-Functional Requirements................................................................................................................8  
3.0. Requirements Specification.......................................................................................................................9  
3.1 External Interface Requirements............................................................................................................9  
3.2 Functional Requirements......................................................................................................................10  
3.2.1 Student Login.................................................................................................................................10  
3.2.2 Change Student Password..............................................................................................................11  
3.2.3 Choose Mess option.......................................................................................................................11  
3.2.4 View Menu.....................................................................................................................................12  
3.2.5 Place NC order...............................................................................................................................12  
3.2.6 Place special order.........................................................................................................................13  
3.2.7 View Statistics................................................................................................................................13  
3.2.8 Give Feedback...............................................................................................................................14  
3.2.9 Swipe Card/ Enter ID.....................................................................................................................14  
3.2.10 Logout.........................................................................................................................................14  
3.2.11 Mess admin login.........................................................................................................................15  
3.2.12 Change Admin Password............................................................................................................15  
3.2.14 View Orders.....................................................................................................................................15  
3.2.8 View Feedback...............................................................................................................................16  
3.3 Detailed Non-Functional Requirements...............................................................................................16  
3.3.1 Logical Structure of the Data.........................................................................................................17  
3.3.2 Security..........................................................................................................................................19  
i  
**List of Figures**Figure 1 - System Environment........................................................................................................................1  
Figure 2 - Logical Structure of the Article Manager Data..............................................................................20  
ii  
**1.0. Introduction  
*1.1. Purpose***The purpose of this document is to present a detailed description of the Mess  
Management System. It will explain the purpose and features of the system,  
the interfaces of the system, what the system will do, the constraints under  
which it must operate and what kind of system interactions take place.  
***1.2. Scope of Project***This software system will be a mess management system which  
consists of two separate softwares, one to be used by the students who eat in  
the mess and one for the mess administrator. The student software allows  
students to login and choose the monthly mess option, place orders for the  
night canteen online, book the mess for special occasions, view the daily  
mess menu and the night canteen (NC) menu, view food consumption  
statistics, inform the mess authorities when he/she won't be eating in the  
mess and submit feedback. The admin software will allow the mess  
administrator to make changes to the daily mess menu, upload food  
consumption statistics and view net profit, view orders for the night mess  
and approve any special lunch orders as well along with reading feedback.  
The system will utilize a relational database for handling all the data such as  
iii  
menu items, prices, student IDs, consumption statistics, student feedback,  
etc.  
***1.3. Glossary***

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Database | Collection of all the information monitored by this system. |
| Mess Admin | The administrator in charge of the mess - updates the databases, uploads statistics, keeps track of orders, resets the system after every meal, etc. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Student | A student avails the mess facilities. |
| Menu | A collection of all the food items which are available in the mess. |
| Order | A collection of food items from the menu that the student wants to buy. |
| Special Order | A list of meal preferences along with other data such as the date and time of the special lunch. A special order will typically be made for a club/department meeting or for special occasions such as festivals. |
| NC | Night Canteen (open from 11:15pm to 2:00 am), sells items which are not on the daily mess menu in a pay-and-eat system. |

iv  
***1.4. References  
1.5. Overview of Document***The next chapter, the Overall Description section, of this document  
gives an overview of the functionality of the product. It describes the  
informal requirements and is used to establish a context for the technical  
requirements specification in the next chapter.  
The third chapter, Requirements Specification section, of this  
document is written primarily for the developers and describes in technical  
terms the details of the functionality of the product.  
Both sections of the document describe the same software product in  
its entirety, but are intended for different audiences and thus use different  
language.  
  
**2.0.Overall Description  
*2.1 System Environment*Figure 1 - System Environment**The Mess Management System has two active actors and one  
relational database.  
The Students use a web portal to which they can login with their ID and  
password and avail all the facilities available. The Mess Admin has access to  
SRS V 1.0 1 October 29, 2014  
a different interface on his/her end, with a login and password. A relational  
database is used to store all the data such as feedback, consumption  
statistics, list of registered students, menus with items and prices, etc.  
***2.2 Functional Requirements Specification***This section outlines the use cases for each of the actors separately.  
The Students have a different set of use cases (which are related to availing  
the mess facilities) than the Mess Admin (which are related to managing the  
various aspects of the mess such as special orders, NC orders, student mess  
option, menu updates, etc.).  
  
2.2.1 Student Use CasesA brief description of all the Student Use Cases:  
1) **Login** – The student logs into the system so that he can carry on with  
other options like choosing mess option or placing orders for the night  
canteen. The student has to enter his ID number and password which  
will be then compared with the database entries to validate his login  
credentials.  
2) **Change Password** – The student changes the current account  
password. To do this he has to retype old password so that his identity  
can be confirmed by matching it with database entires and after it has  
been validated, he has to enter a new password, which will then be  
updated in the database.  
3) **Choose Mess Option** – The student chooses his monthly mess option,  
A or C mess. His choice is then sent to the database where it is  
updated. This option is only available for fixed dates in a month and  
cannot be changed on any other days. If no option is chosen, one will  
be randomly selected.  
4) **View Menu** – The student gets to view the menu for the week. He can  
also choose to view a particular menu like Monday's lunch menu or  
the Night mess menu. The query from the student is sent to the  
database where the data is read and is sent back to the student to be  
displayed on the screen.  
5) **View Statistics** – The student's query for Statistics is sent to the  
database on the server where the relevant data is read and sent back to  
the student. The statistics include the food consumption and can be  
viewed in a daily, weekly or monthly format and shows what percent  
of people ate and how much food was consumed and wasted.  
6) **Book NC order** – The student can view the night mess menu and  
choose items from it to preorder online. This saves a lot of time and is  
more efficient than standing in a queue. The student's order is sent to  
the database where it is processed by the system. The student can also  
view the current order status and collect it when it is ready.  
7) **Book Special Order** – This option is meant for clubs and departments  
or any group of students who want to book the mess for a particular  
time and date and have a special lunch or dinner. It involves  
submitting the requirements of the order to the mess admin who has to  
approve it and provide a quotation for the amount. The system has to  
validate the order by checking if the mess hasn't already been booked  
for that time.  
8) **Swipe Card / Enter ID** – When the student goes to the mess, he has  
to either swipe the card or enter his ID. The Id entered will be looked  
up in the database to check if the student has registered for that mess  
and if he has already eaten or not. This will be reset after each meal. A  
failure or success message will be displayed based on the validation  
result.  
9) **Logout** – The student logs out of the system so that nobody else can  
modify his mess option or book orders from his account.  
  
2.2.2 Mess Admin Use Cases

A brief description of the Mess Admin Use Cases:  
1) **Login** – The administrator has to login to the system so that he can  
carry on with the other activities such as updating the mess menu,  
approving special orders, viewing and uploading statistics, viewing  
NC orders etc. The data entered will be matched with the fields in the  
database to validate the admin's credentials.  
2) **Change Password** – The admin changes the current account  
password. To do this he has to retype old password so that his identity  
can be confirmed by matching it with database entires and after it has  
been validated, he has to enter a new password, which will then be  
updated in the database.  
3) **Update Menu** – The administrator can update the menu for any meal  
for any day of the week or the might mess menu by selecting new  
food items to add to the menu or by removing unpopular food items.  
The changes after the updation are saved in the database so that the  
students can see the updated menu.  
4) **View NC Orders** – The admin can check the number of orders  
received for the night mess and also pass along that to the kitchen staff  
so that they can prepare the meals according to those specifications.  
The admin can then update the order status in the database to  
completed so that the student can come and pick it up after which its  
status will be delivered.  
5) **Approve Special Orders** – The admin can view the Special orders  
submitted by a group of students, provide a quotation for the special  
meal and if that is acceptable to the group, he can approve the order so  
that on that day, food for the special order will be prepared for that  
group of students. These changes will be updated in the database so  
that the student can view the order status and total cost.  
6) **View Feedback** – The mess admin can view the feedback submitted  
by the students. The database will be queried for feedback and  
complaints and the admin can check all these forms and take the  
required action on them. He can also update the feedback status as  
read or the complaint status as per its current state.  
7) **Reset System** – The mess admin can trigger a hard reset of the system  
where the list of people who have eaten their meal is erased. This will  
typically occur automatically after every meal and so this hard reset  
will seldom be used.  
8) **Logout –** The mess admin can log out safely while he/she is not using  
the software in order to avoid any misuse.  
***2.3 User Characteristics***The Students and Mess Admin are expected to have a basic working  
knowledge of a computer in order to use this software. The User  
Interface will be quite intuitive, so any advanced knowledge will not be  
necessary.  
***2.4 Non-Functional Requirements***The physical machine to be used in the mess needs to have internet  
access in order to connect to the database. This software will not assume that  
a code scanner hardware is available on the machine, and so the ID input  
will be done via keyboard. Students need internet access on their devices as  
well, since all the data will be stored on the database which the software will  
need to connect to.

**3.0. Requirements Specification  
*3.1 External Interface Requirements***3.11 User Interfaces: The Interface will be in the form of a web app.  
It is designed to be functional and minimal in its styling. All  
options will be displayed in a menu-based format. HTML and CSS  
will be used to setup the page layout and add minimal styling to  
make the interface user friendly.  
3.12 Hardware Interfaces: A webserver will be required so that the  
students and the mess admin can connect to it to exchange  
information. The server has a database to store all the data  
entries. The Server will have to have a highspeed 1 Gigabit  
ethernet connection to the college's local network.  
3.13 Software Interfaces: The server will be hosted using Apache  
Tomcat Webserver (Version 8.0.14). It will also have a MySQL  
relational database. The main backend processing will be done  
using Java Server Pages (JSP) including connecting to and  
accessing the database and processing requests.  
3.14 Communications Interfaces: The main communication protocol  
will be Hyper Text Transfer Protocol (HTTP). This will be used to  
transfer information back and forth from the client to the server.  
HTTP GET and POST will be used to send the information.

***3.2 Functional Requirements***3.2.1 Student Login

|  |  |
| --- | --- |
| **Use Case Name** | Login |
| **Trigger** | The Student clicks on the login button on the login page |
| **Precondition** | The Student has entered his login details on the login page |
| **Basic Path** | 1. The student navigates to the login page 2. The student enters the username and password 3. The student clicks the login button 4. If the form data is empty, system shows a prompt for login details 5. If data is not empty, it is sent to the server 6. The server compares the login data with the password stored in the database 7. If login credentials are verified, the student is logged in 8. If not, the student is prompted to enter the login details again |
| **Postcondition** | The Student is logged in and is taken to his account page |
| **Exception Paths** | The Student may terminate the login at any time. |

3.2.2 Change Student Password

|  |  |
| --- | --- |
| **Use Case Name** | Change Password |
| **Trigger** | The student clicks on the change password button on his account page |
| **Precondition** | The student has logged in and is on his account page |
| **Basic Path** | 1. The student enters the previous password and the new desired password. 2. The data is sent to the server if it is not empty 3. The server validates the user's password and then updates his password in the database |

|  |  |
| --- | --- |
| **Alternative Paths** | The student may click on the forgot password link on the login page to have his password changed after having it emailed to him |
| **Postcondition** | The user's password is updated in the database |
| **Exception Paths** | The attempt may be abandoned at any time. |

3.2.3 Choose Mess option

|  |  |
| --- | --- |
| **Use Case Name** | Choose Mess option |
| **Trigger** | The user clicks the choose mess option button on his account page |
| **Precondition** | The user has logged in and the date is a valid date for selecting mess option (since it can only be done in a particular time of the month) |
| **Basic Path** | 1. The student is given two options in the form of radio button to choose from 2. Once the user has selected an option, he clicks the submit button where the server receives this data 3. The server updates his mess choice in the database and also updates that he has chosen an option so that he cannot change it again |
| **Alternative Paths** | If the student clicks submit without selecting an option, the system will prompt for an option |
| **Postcondition** | The student's mess choice for the following month is updated in the database |
| **Exception Paths** | The student may abandon the operation at any time. |
| **Other** | Date and time validity needs to be checked. Also if no option is selected in that time period, the system will automatically assign a random choice |

3.2.4 View Menu

|  |  |
| --- | --- |
| **Use Case Name** | View Menu |
| **Trigger** | The student clicks the view menu button |
| **Precondition** | The student has clicked the view menu button on the home page of the mess management system |
| **Basic Path** | 1. The student can choose which day's menu he wants to view 2. The choice is sent to the server where the database is queried for that particular menu 2. 3. The server sends a response back to the client with the data about the menu 4. The response XML is parsed and formatted into an easy to read menu format |
| **Alternative Paths** | The student can also choose to view the night mess menu along with the cost of the food items |
| **Postcondition** | The Menu is displayed on the screen |
| **Exception Paths** | The student may abandon the operation at any time. |
| **Other** | Menu consists of the menu items and their details |

3.2.5 Place NC order

|  |  |
| --- | --- |
| **Use Case Name** | Place NC order |
| **Trigger** | The student clicks the place NC order button on his account page |
| **Precondition** | The student has logged in |
| **Basic Path** | 1. The System displays the NC menu 2. The student selects which items he wants to add to order 3. The he chooses the quantity and can add more items or checkout 4. The order details are sent back to the server which stores them in the database 5. Order status is updated on the screen |
| **Alternative Paths** | The student can go back and edit his choices or preferences at any time before checking out but not once the order has been placed. He can also check order status |
| **Postcondition** | The database has been updated with the student's order. |
| **Exception Paths** | The user may abandon the order at any time before submitting |

3.2.6 Place special order

|  |  |
| --- | --- |
| **Use Case Name** | Place special order |
| **Trigger** | The student clicks the place special order button on his account page |
| **Precondition** | The student has logged in and is on his account page |
| **Basic Path** | 1.The student is presented with a text box where he has to specify the details of his special order 2. The form is submitted to the server which stores it in the database 3. The user can then check periodically to view the order status |
| **Postcondition** | The student's special order has been submitted |
| **Exception Paths** | If the booking dates and times are clashing in the database, then the system informs the student to pick another date |

3.2.7 Give Feedback

|  |  |
| --- | --- |
| **Use Case Name** | Submit Feedback |
| **Trigger** | The student clicks the give feedback button on his account page |
| **Precondition** | The student has logged in to his account page |
| **Basic Path** | 1. Student is presented with a text form to write his feedback 2. The feedback is submitted to the server where it is stored in the database |
| **Postcondition** | The database has been updated with the student's feedback |
| **Exception Paths** | The Student may abandon the operation at any time. |
| **Other** | The feedback also has a status associated with it so that the user can track if it has been acted upon or not |

3.2.8 Swipe Card/ Enter ID

|  |  |
| --- | --- |
| **Use Case Name** | Swipe card / Enter ID |
| **Trigger** | The Editor has selected to check status of all active articles. |
| **Precondition** | The mess admin has accessed the student entry page |
| **Basic Path** | 1. There is a text box where the student has to type his ID number. 2. The ID is sent to the server where it checks if the student has opted for this mess and if he has already eaten 3. The student's details like name and photo are shown on screen and he is allowed to take a plate |
| **Alternative Paths** | If authentication fails, the system will show an error message |
| **Postcondition** | The database is updated to reflect the changes made |
| **Exception** | Invalid ID's can cause exceptions |

|  |
| --- |
| **Paths** |
| **Other** | The Mess admin has to take care the the photo shown by the sytem matches the student or else he may log in with false credentials |

3.2.9 Logout

|  |  |
| --- | --- |
| **Use Case Name** | Logout |
| **Trigger** | The Student clicks the logout button |
| **Precondition** | The Student has already logged in |
| **Basic Path** | 1. The student is redirected to the login page 2. The server ends the student's current session and logs him out of the system |
| **Alternative Paths** | None. |
| **Postcondition** | The Student is logged out of the system |
| **Exception Paths** | If the student has logged in from multiple devices, there might be a conflict |
| **Other** | None |

3.2.10 Mess admin login

|  |  |
| --- | --- |
| **Use Case Name** | Login |
| **Trigger** | The Admin clicks on the login button on the admin login page |
| **Precondition** | The admin has entered his login details on the login page |
| **Basic Path** | 1. The admin navigates to the login page 2. The admin enters the username and password 3. The admin clicks the login button 4. If the form data is empty, system shows a prompt for login detail 5. If data is not empty, it is sent to the server 6. The server compares the login data with the password stored in the database 7. If login credentials are verified, the student is logged in 8. If not, the admin is prompted to enter the login details again |
| **Postcondition** | The admin is logged in and is taken to his account page |
| **Exception Paths** | The admin may terminate the login at any time. |

3.2.11 Change Admin Password

|  |  |
| --- | --- |
| **Use Case Name** | Change Password |
| **Trigger** | The admin clicks on the change password button on his account page |
| **Precondition** | The admin has logged in and is on his account page |
| **Basic Path** | 4. The admin enters the previous password and the new desired password. 5. The data is sent to the server if it is not empty 6. The server validates the admin's password and then updates his password in the database |
| **Alternative Paths** | The admin may click on the forgot password link on the login page to have his password changed after having it emailed to him |
| **Postcondition** | The admin's password is updated in the database |
| **Exception Paths** | The attempt may be abandoned at any time. |

3.2.12

|  |  |
| --- | --- |
| **Use Case Name** | Update Menu |
| **Trigger** | The admin clicks the update menu button on his account page |
| **Precondition** | The admin has logged into his account |
| **Basic Path** | 1. The admin gets a screen of the current menu 2. He can then create new items which will be added to the database 3. Using these food items, the admin can make changes to the men 4. The changes made will be sent to the server 5.The server reflects these changes by updating the menu database |
| **Alternative Paths** | The admin can also choose to edit the Night mess menu in the same way |
| **Postcondition** | The updated menu is saved in the database |
| **Exception Paths** | The admin may abandon the operation at any time in which case no changes will be made. |
| **Other** | Menu consists of the menu items and their details |

*3.2.13 View Orders*

|  |  |
| --- | --- |
| **Use Case Name** | View NC order, View Special orders |
| **Trigger** | The admin clicks the view orders button on his account page |
| **Precondition** | The admin has logged in |
| **Basic Path** | 1. The admin sends a query to the database for a list of the orders 2. The list is sent from the server back to the admin 3. The admin can then pass on the order information to the kitchen staff 4. Once the order is completed the admin can mark the order status as complete or delivered |
| **Alternative Paths** | The admin can view the special orders in the same way and approve them similarly |
| **Postcondition** | The Order queue is updated along with the order status in the database |
| **Exception Paths** | The admin may abandon the operation at any time. |
| **Other** | The changes made in the database will allow the students to check order status and collect their food accordingly |

3.2.14 View Feedback

|  |  |
| --- | --- |
| **Use Case Name** | View Feedback |
| **Trigger** | The admin clicks the view feedback button on his account page |
| **Precondition** | The admin has logged in to his account page |
| **Basic Path** | 1. The server reads the database and send over a list of the feedback forms to the admin 2. The admin can read each one of them and then mark them as read, unread or to be acted upon 3. These changes are then sent to the database |
| **Postcondition** | The database has been updated with the feedback status |
| **Exception Paths** | The admin may abandon the operation at any time. |
| **Other** | The feedback status will be used to store all unread forms and only a the most recent ones which have been acted upon |

3.2.15 Reset System

|  |  |
| --- | --- |
| **Use Case Name** | Reset System |
| **Trigger** | The admin clicks the reset system button on his account page |
| **Precondition** | The admin has logged in to his account page |
| **Basic Path** | 1. The admin's command is relayed to the server 2. The server deletes the record of students who have eaten in the mess for that meal from the database |
| **Postcondition** | The database entries of status of students are deleted from the system |
| **Exception Paths** | This operation can cause issues since wiping data manually is not recommended |
| **Other** | This option should only be used in special circumstances since the system automatically resets after each meal |

3.2.16 Logout

|  |  |
| --- | --- |
| **Use Case Name** | Logout |
| **Trigger** | The admin clicks the logout button |
| **Precondition** | The admin has already logged in |
| **Basic Path** | 1. The admin is redirected to the login page 2. The server ends the admin's current session and logs him out of the system |
| **Alternative Paths** | None. |
| **Postcondition** | The admin is logged out of the system |
| **Exception Paths** | None |
| **Other** | None |

***3.3 Detailed Non-Functional Requirements***3.3.1 Logical Structure of the Data  
Data Entries in the Database  
**Student Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Student |  |
| Email Address | Text | BITS Email Id of student |  |
| Mess Option | Text | A or C mess |  |
| ID number | Text | BITS ID number |  |
| Password | Text | Password of student | Encrypted form |
| Has Eaten | Boolean | Whether student has eaten in mess | Resets automatically after each meal |

**Admin Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of the Admin |  |
| Password | Text | Password of Admin | Encrypted form |
| Email Address | Text | Email Id of Admin |  |

**Menu Item Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Item |  |
| Type | Text | Regular or NC |  |
| Price | Number | Price of NC menu item | Optional |

**Menu Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Day | Text | Day of the week |  |
| Meal | Text | Lunch, Dinner etc |  |
| Type | Text | Regular or NC menu |  |

**Order Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ItemList | Text | A list of item ordered |  |
| Quantity | Text | A list of quantities |  |
| Status | Text | The order status |  |
| Student ID | Text | ID of student who ordered it |  |
| Order number | Number | A sequence number |  |

**Feedback Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Content | Text | Feedback content |  |
| Status | Text | Current feedback status | Read, unread or processed |
| Student ID | Text | ID of student |  |

3.3.2 Security  
The server on which the Mess Management resides will have its own  
security to prevent unauthorized *write*/*delete* access. There is no restriction  
on *read* access. In case a password is forgotten, a new one will be emailed to  
that user's email ID. In addition, the passwords will be MD5 hashed forsecurity. An automatic logout system will log out a student after 10 minutes.