**Initial Functional Tests**

**Use Case: MRM Startup**

**Function Being Tested**: Machine is started when the switch is turned "on"

**Initial System State**: Machine is off

**Input**: Activate the "on" switch

**Expected Output**: Machine asks the operator for verification

**Function Being Tested**: Verification process of operator

**Initial System State**: Machine requesting to enter the verification

**Input**: Entered wrong verification id

**Expected Output**: Machine showing invalid credential and asks for valid id

**Function Being Tested**: Verification process of operator

**Initial System State**: Machine requesting to enter the verification

**Input**: Entered correct verification id

**Expected Output**: Machine asks to enter the total inventory

**Function Being Tested**: Inventory setup

**Initial System State**: Machine asks to enter the total inventory

**Input**: Inventory entered

**Expected Output**: Machine is on and displaying message to swipe customer credit card

**Use Case: Verification**

**Function Being Tested**: Credit Card verification

**Initial System State**: Machine requesting for customer credit card

**Input**: Insert an unreadable card

**Expected Output**: Card is ejected; System displays an error screen; System is ready to start a new session

**Function Being Tested**: Credit Card verification

**Initial System State**: Machine requesting for customer credit card

**Input**: Insert a readable card

**Expected Output**: Machine asks to enter the zip code

**Function Being Tested**: Zip Code verification

**Initial System State**: Machine asks to enter Zip code

**Input**: Zip Code entered wrong 3 times

**Expected Output**: Cancel transaction

**Function Being Tested**: Zip Code verification

**Initial System State**: Machine asks to enter Zip code

**Input**: correct zip Code entered

**Expected Output**: Continue session; allowing customer to select option from menu if they want to rent a movie or if they want to deposit

**Use Case: Movie Search**

**Function Being Tested**: Search for non existed movies

**Initial System State**: Machine asks to enter movie name

**Input**: Entered random name

**Expected Output**: Machine display error that there is no movie with this name, enter name correctly

**Function Being Tested**: Search for unavailable movie

**Initial System State**: Machine asks to enter movie name

**Input**: Entered movie name which is not currently available

**Expected Output**: Machine display message that movie currently not available; send the movie name to server database of unavailable movies; asks customer to reserve movie

**Function Being Tested**: Search for available movie

**Initial System State**: Machine asks to enter movie name

**Input**: Entered movie name which is currently available

**Expected Output**: Machine display movie name with its cost; asks customer to put the movie in shopping cart

**Use Case: Reserve movie**

**Function Being Tested**: Allow user to not reserve the movie

**Initial System State**: Machine asks customer if they want to reserve the movie

**Input**: Select “no” option for not reserving the movie

**Expected Output**: Machine doesn’t reserve the movie and display menu to search for any other movie

**Function Being Tested**: Allow user to reserve the movie

**Initial System State**: Machine asks customer if they want to reserve the movie

**Input**: Select “yes” option to reserve the movie

**Expected Output**: Machine will reserve the movie and asks customer to enter their mail id to inform them on movie arrival

**Use Case: Rent movie**

**Function Being Tested**: Allow user to deny the renting of the movie

**Initial System State**: Machine asks customer if they want to rent the movie

**Input**: Select “no” option for not renting the movie

**Expected Output**: Machine will not disperse that movie and asks customer to enter another movie name

**Function Being Tested**: Allow user to rent the movie

**Initial System State**: Machine asks customer if they want to rent the movie

**Input**: Select “yes” option to rent the movie

**Expected Output**: Machine will disperse that movie; send log to the server; hold customer credit card information and asks customer if they want to continue the session

**Use Case: Deposit movie**

**Function Being Tested**: Deposit movie menu

**Initial System State**: Machine asks customer if they want to deposit movie

**Input**: Select “deposit” option to deposit the movie

**Expected Output**: Machine asks customer to deposit the movie

**Function Being Tested**: Deposit wrong movie

**Initial System State**: Machine expecting customer to deposit movie

**Input**: Deposit some other unrented movie

**Expected Output**: Machine checks database; eject the deposit movie; asks to deposit the collected movie

**Function Being Tested**: Deposit correct movie

**Initial System State**: Machine expecting customer to deposit movie

**Input**: Deposit movie

**Expected Output**: Machine checks database; collect the movie; remove hold on customer’s credit card; send the log to the server

**Use Case: Cancel operation**

**Function Being Tested**: Abort the operation after confirming the order and transaction processing

**Initial System State**: Machine has confirmed the payment

**Input**: Select “Cancel” button

**Expected Output**: Machine warn user that the operation can’t be aborted at this stage

**Function Being Tested**: Abort the operation in between any state of machine

**Initial System State**: Machine can be in any stage before making payment

**Input**: Select “Cancel” button

**Expected Output**: Machine confirms the customers if they are sure if they want to cancel the operation; on selecting “yes” machine will return back the credit card to the customer

**Use Case: Error message**

**Function Being Tested**: Proper display of error message

**Initial System State**: Machine working correctly in between any state

**Input**: Select any wrong operation like entered wrong movie name or deposit wrong movie

**Expected Output**: Machine throws correct error message to the customer

**Use Case: Log Entry**

**Function Being Tested**: Server maintaining the log history

**Initial System State**: Machine is ON

**Input**: Perform any transaction, reserve a movie or any other operation

**Expected Output**: Machine is sending all the operation logs correctly to the server

**Function Being Tested**: Machine maintaining the log history

**Initial System State**: Machine is ON

**Input**: Perform any transaction, reserve a movie or any other operation

**Expected Output**: Machine is maintaining a replica of the log history which is maintained by server

**Use Case: MRM Stop**

**Function Being Tested**: Machine shut downs in between servicing a customer

**Initial System State**: Machine is ON

**Input**: Push machine STOP button

**Expected Output**: Machine won’t shut down asking to complete the customer service first

**Function Being Tested**: Enter movie while machine is in running mode

**Initial System State**: Machine is ON

**Input**: Reload new movie

**Expected Output**: Machine won’t allow operator to add movies while in running mode

**Function Being Tested**: Remove movie while machine is in running mode

**Initial System State**: Machine is ON

**Input**: Remove a movie

**Expected Output**: Machine won’t allow operator to remove movies while in running mode

**Function Being Tested**: Shutdown machine while not in service

**Initial System State**: Machine is ON

**Input**: Push STOP button

**Expected Output**: Machine shut down and allow operator to add new movies and remove any movie