



مواصفات المشروع

## Payment Application

### Development environment preparation

المعايير	يفي بالمواصفات
Create modules folders	<p>1. Create a new project</p> <p>2. Create "<b>Application</b>" folder</p> <p>3. Create "<b>Card</b>" folder</p> <p>4. Create "<b>Terminal</b>" folder</p> <p>5. Create "<b>Server</b>" folder</p> <p>Note: To create a folder in Microsoft Visual Studio</p> <p>1. In the solution explorer, right-click on the project name</p> <p>2. Go to Add</p> <p>3. Select Folder</p> <p>4. Give a name to that folder</p> <p>You should deliver a screenshot of the solution explorer that clarifies your folder structure</p>
Create .c and .h file for each module	<p>1. In the "<b>Application</b>" folder create <b>app.c</b> and <b>app.h</b> files</p> <p>2. In the "<b>Card</b>" folder create <b>card.c</b> and <b>card.h</b> files</p> <p>3. In the "<b>Terminal</b>" folder create <b>terminal.c</b> and <b>terminal.h</b> files</p>

المعايير	يفي بالمواصفات
<p>In the "Server" folder create <b>server.c</b> and <b>server.h</b> files .4</p> <p>Note: To create a file into a folder in Microsoft Visual Studio</p> <p>In the solution explorer, right-click on the folder you .1 want Go to Add .2 Select New Item .3 Select file type, .cpp or .h .4 If a .cpp is chosen, change the extension to .c .5 "Give a name to that file .6</p> <p>You should deliver a screenshot of the solution explorer .that clarifies files in each folder</p>	
<p>In the <b>app.h</b> file add the header file guard .1 In the <b>card.h</b> file add the header file guard .2 In the <b>terminal.h</b> file add the header file guard .3 In <b>server.h</b> file add the header file guard .4</p> <p>You should deliver a screenshot for each .h file, the file name must appear in the screenshot, and the header file guard</p>	<p>Add header file gaurd</p>

Implement the card module

المعايير	يفي بالمواصفات
<p>1. Use the following typedef as-is :</p> <pre>typedef struct ST_cardData_t {     ;uint8_t cardHolderName[25]     ;uint8_t primaryAccountNumber[20]     ;uint8_t cardExpirationDate[6]     ;ST_cardData_t{</pre> <pre>typedef enum EN_cardError_t {     CARD_OK, WRONG_NAME, WRONG_EXP_DATE,     WRONG_PAN     ;EN_cardError_t{</pre> <p>2. Use the following prototypes as is :</p> <pre>EN_cardError_t getCardHolderName(ST_cardD     ;ata_t *cardData) EN_cardError_t getCardExpiryDate(ST_cardD     ;ata_t *cardData) EN_cardError_t getCardPAN(ST_cardData_t *     ;cardData)</pre> <p>You should deliver a screenshot of your card.h file</p>	<p>Fill in card.h file with functions' prototypes and typedefs</p>
<p>1. This function will ask for the cardholder's name .and store it into card data</p> <p>2. Cardholder name is 24 alphabetic characters .string max and 20 min</p> <p>3. If the cardholder name is <code>NULL</code> , less than 20 characters or more than 24 will return a <code>WRONG_NAME</code> error, else return <code>CARD_OK</code></p> <p>4. :Test your function</p>	<p>Implement getCardHolderName function</p>

المعايير	يفي بالمواصفات
	<p>Create a test function ◦</p> <pre>void getCardHolderNameTest(void);</pre> <p>to test all possible scenarios, happy-case, and worst-case scenarios</p> <p>Print all results of your test cases on the console window, use the following as a guide</p> <pre> Tester Name: your name Function Name: getCardHolderName :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n :Input Data :Expected Result :Actual Result </pre> <p>You should deliver the test function as well as a .5 screenshot of the results on the console</p>
<p>This function will ask for the card expiry date .1 and store it in card data</p> <p>Card expiry date is 5 characters string in the .2 "format "MM/YY", e.g "05/25</p> <p>If the card expiry date is <code>NULL</code> , less or more .3 than 5 characters, or has the wrong format will</p>	<p>Implement getCardExpiryDate function</p>

المعايير	يفي بالمواصفات
<p>return the <code>WRONG_EXP_DATE</code> error, else return <code>CARD_OK</code>.</p> <p>4. Test your function.</p> <p>Create a test function</p> <ul style="list-style-type: none"> <li>void <code>getCardExpiryDateTest (void);</code> to test all possible scenarios, happy-case, and worst-case scenarios</li> <li>Print all results of your test cases on the console window, use the following as a guide</li> </ul> <pre> Tester Name: your name Function Name: getCardExpiryDate :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n :Input Data :Expected Result :Actual Result </pre> <p>5. You should deliver the test function as well as a screenshot of the results on the console.</p>	
<p>1. This function will ask for the card's <b>Primary Account Number</b> and store it in card data</p> <p>2. PAN is 20 numeric characters string, 19 character max, and 16 character min</p>	<p>Implement <code>getCardPAN</code> function</p>

المعايير	يفي بالمواصفات
	<p>If the PAN is <code>NULL</code>, less than 16 or more than 19 characters, will return the <code>WRONG_PAN</code> error, else <code>.return CARD_OK</code></p> <p>4. Test your function</p> <p>Create a test function</p> <ul style="list-style-type: none"> <li>◦ <code>void getCardPANTest (void);</code> to test all possible scenarios, happy-case, and worst-case scenarios</li> <li>◦ Print all results of your test cases on the console window, use the following as a guide</li> </ul> <pre> Tester Name: your name Function Name: getCardPAN :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n :Input Data :Expected Result :Actual Result </pre> <p>5. You should deliver the test function as well as a screenshot of the results on the console</p>
<p>1. Record a video where you discuss each function you implemented in this module</p> <p>2. Explain and execute all test functions you made</p> <p>3. The video is <b>4 minutes maximum</b></p>	<p>Explain your work</p>

المعايير	يفي بالموصفات
	<p>.4 You may record it in Arabic or English</p> <p>.5 Muted videos will not be acceptable</p> <p>.6 All of the above are mandatory to pass this criterion</p>

### Implement the terminal module

المعايير	يفي بالموصفات
<p>Fill in terminal.h file with functions' prototypes and typedefs</p>	<p>1. Use the following typedef as is</p> <pre>typedef struct ST_terminalData_t {     ;float transAmount     ;float maxTransAmount     ;uint8_t transactionDate[11]     ;ST_terminalData_t{</pre> <pre>typedef enum EN_terminalError_t {     TERMINAL_OK, WRONG_DATE, EXPIRED_CARD,     INVALID_CARD, INVALID_AMOUNT, EXCEED_MAX_AM     OUNT, INVALID_MAX_AMOUNT     ; EN_terminalError_t{</pre> <p>2. Use the following prototypes as is</p> <pre>EN_terminalError_t getTransactionDate(ST_te     ;rminalData_t *termData) EN_terminalError_t isCardExpired(ST_cardDat     a_t *cardData, ST_terminalData_t *termDat     ;a) EN_terminalError_t getTransactionAmount(ST_     ;terminalData_t *termData) EN_terminalError_t isBelowMaxAmount(ST_term     ;inalData_t *termData) EN_terminalError_t setMaxAmount(ST_terminal</pre>

المعايير	يفي بالمواصفات
	<pre> ;Data_t *termData, float maxAmount) EN_terminalError_t isValidCardPAN(ST_cardDa ta_t *cardData); // Optional </pre> <p>You should deliver a screenshot for your terminal.h.3 file</p>
<p>This function will ask for the transaction date and .1          .store it in terminal data          Transaction date is 10 characters string in the .2          .format DD/MM/YYYY, e.g 25/06/2022          If the transaction date is <code>NULL</code> or is less than 10 .3          characters or wrong format will return the          . <code>WRONG_DATE</code> error, else return <code>TERMINAL_OK</code>          :Test your function .4</p> <p>Create a test function ◦</p> <pre>void getTransactionDateTest(void);</pre> <p>to test all possible scenarios, happy-          .case, and worst-case scenarios          Print all results of your test cases on ◦          the console window, and use the          :following as a guide</p> <pre> Tester Name: your name Function Name: getTransactionDate :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n </pre>	<p>Implement          getTransactionDate          function</p>



المعايير	يفي بالمواصفات
	<div data-bbox="342 222 855 378"> :Input Data  :Expected Result  :Actual Result </div> <p data-bbox="792 422 946 453">.5 Optional</p> <p data-bbox="280 468 919 583">The function will read the current date from your computer and store it into terminal data with the .mentioned size and format</p> <p data-bbox="302 600 946 674">You should deliver the test function as well as a .6 screenshot of the results on the console</p>
<p data-bbox="264 821 946 894">This function <b>compares the card expiry date with .1 the transaction date</b></p> <p data-bbox="305 909 946 1031">If the card <b>expiration date is before the .2</b> transaction date will return <code>EXPIRED_CARD</code> , else .return <code>TERMINAL_OK</code></p> <p data-bbox="675 1041 946 1073">.3 Test your function</p> <ul data-bbox="354 1121 894 1423" style="list-style-type: none"> <li>◦ Create a test function <code>void isCardExpriedTest(void);</code> to test all possible scenarios, happy-case, and .worst-case scenarios</li> <li>◦ Print all results of your test cases on the console window, and use the :following as a guide</li> </ul> <div data-bbox="342 1430 855 2026"> <pre> Tester Name: your name Function Name: isCardExpried :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . .</pre> </div>	<p data-bbox="1040 821 1336 894">Implement isCardExpried function</p>

المعايير	يفي بالمواصفات
	<div data-bbox="341 224 855 422"> <pre> :Test Case n :Input Data :Expected Result :Actual Result </pre> </div> <p data-bbox="302 464 948 541">You should deliver the test function as well as a .4 .screenshot of the results on the console</p>
<p data-bbox="251 699 948 777">This function asks for the transaction amount and .1 .saves it into terminal data</p> <p data-bbox="251 787 948 955">If the transaction amount is less than or equal to 0 .2 will return INVALID_AMOUNT , else return TERMINAL_OK . :Test your function .3</p> <p data-bbox="574 1003 893 1035">Create a test function ◦</p> <div data-bbox="341 1045 855 1085"> <pre>void getTransactionAmountTest(void);</pre> </div> <p data-bbox="383 1094 855 1171">to test all possible scenarios, happy- .case, and worst-case scenarios</p> <p data-bbox="383 1182 893 1304">Print all results of your test cases on ◦ the console window, and use the :following as a guide</p> <div data-bbox="341 1310 855 2011"> <pre> Tester Name: your name Function Name: getTransactionAmount :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n </pre> </div>	<p data-bbox="1036 699 1336 821">Implement getTransactionAmount function</p>

المعايير	يفي بالمواصفات
	<div data-bbox="341 222 855 378"> :Input Data  :Expected Result  :Actual Result </div> <p data-bbox="302 422 948 499">You should deliver the test function as well as a .4 .screenshot of the results on the console</p>
<p data-bbox="280 638 948 716">This function <b>compares the transaction amount .1</b> <b>.with the terminal max allowed amount</b></p> <p data-bbox="347 726 948 804">If the transaction amount is <b>larger than the .2</b> <b>terminal max allowed amount</b> will return</p> <p data-bbox="321 814 915 852">. <code>EXCEED_MAX_AMOUNT</code> , else return <code>TERMINAL_OK</code></p> <p data-bbox="672 863 948 900">:Test your function .3</p> <p data-bbox="574 942 894 980">Create a test function ◦</p> <p data-bbox="358 989 855 1108"><code>void isBelowMaxAmountTest(void);</code> to test all possible scenarios, happy-case, .and worst-case scenarios</p> <p data-bbox="386 1119 894 1239">Print all results of your test cases on ◦ the console window, and use the :following as a guide</p> <div data-bbox="341 1249 855 1942"> <pre> Tester Name: your name Function Name: isBelowMaxAmount :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n :Input Data </pre> </div>	<p data-bbox="1073 638 1338 758">Implement isBelowMaxAmount function</p>

المعايير	يفي بالمواصفات
	<div data-bbox="341 224 855 333"> :Expected Result :Actual Result </div> <p>You should deliver the test function as well as a .4 .screenshot of the results on the console</p>
<p>Implement setMaxAmount function</p>	<p>This function <b>takes the maximum allowed amount .1</b> <b>.and stores it into terminal data</b> .Transaction max amount is a <b>float number .2</b> If transaction max amount <b>less than or equal to 0 .3</b> will return the <code>INVALID_MAX_AMOUNT</code> error, else .return <code>TERMINAL_OK</code> .4 :Test your function</p> <p>Create a test function ◦ <code>void setMaxAmountTest(void);</code> to test all possible scenarios, happy-case, and .worst-case scenarios</p> <p>Print all results of your test cases on ◦ the console window, and use the :following as a guide</p> <div data-bbox="341 1247 855 1923"> <pre> Tester Name: your name Function Name: setMaxAmount :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n :Input Data </pre> </div>

المعايير	يفي بالمواصفات
	<div data-bbox="341 222 855 333"> :Expected Result :Actual Result </div> <p data-bbox="302 375 946 451">You should deliver the test function as well as a .5 .screenshot of the results on the console</p>
<p data-bbox="341 592 946 667">This function will <b>check if the PAN is a Luhn .1</b> <b>.number or not</b></p> <p data-bbox="341 680 946 756">For more about <b>Luhn number</b>, and how to .2 .generate and check please refer to this <a href="#">Site</a></p> <p data-bbox="326 768 946 844">If the number is not Luhn number, will return .3 . <code>INVALID_CARD</code> , else will return <code>TERMINAL_OK</code></p> <p data-bbox="672 856 946 894">.4 :Test your function</p> <p data-bbox="574 938 893 970">Create a test function ◦</p> <p data-bbox="355 982 855 1104"><code>void isValidCardPANTest(void);</code> to test all possible scenarios, happy-case, .and worst-case scenarios</p> <p data-bbox="386 1117 893 1239">Print all results of your test cases on ◦ the console window, and use the :following as a guide</p> <div data-bbox="341 1247 855 1940"> <pre> Tester Name: your name Function Name: isValidCardPAN :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n :Input Data </pre> </div>	<p data-bbox="1083 592 1336 714">Implement isValidCardPAN function (<b>Optional</b>)</p>

المعايير	يفي بالمواصفات
<div data-bbox="341 226 855 333"> :Expected Result :Actual Result </div> <p data-bbox="266 375 946 499">If you are going to implement this function, please .5 deliver the test function as well as a screenshot of .the results on the console</p>	
<p data-bbox="293 638 946 982">Record a video where you discuss each function .1 .you implemented in this module .2.Explain and execute all test functions you made .3.The video is <b>4 minutes maximum</b> .4.You may record it in Arabic or English .5.Muted videos will not be acceptable .6.All of the above are mandatory to pass this .criterion</p>	<p data-bbox="1101 638 1336 674">Explain your work</p>

### Implement the server module

المعايير	يفي بالمواصفات
<p data-bbox="505 1354 935 1390">.1 Use the following typedef as-is</p> <div data-bbox="207 1392 907 1703"> <pre data-bbox="240 1423 885 1675">typedef enum EN_transState_t {     APPROVED, DECLINED_INSUFFECIENT_FUND, DE     CLINED_STOLEN_CARD, FRAUD_CARD, INTERNAL_SERV     ER_ERROR };EN_transStat_t{</pre> </div> <div data-bbox="207 1726 907 2011"> <pre data-bbox="370 1757 885 1969">typedef struct ST_transaction_t {     ;ST_cardData_t cardHolderData     ;ST_terminalData_t terminalData     ;EN_transState_t transState</pre> </div>	<p data-bbox="1045 1354 1336 1478">Fill in server.h file with functions' prototypes and typedefs</p>

المعايير	يفي بالموصفات
	<pre> ;uint32_t transactionSequenceNumber ;ST_transaction{  typedef enum EN_serverError_t } SERVER_OK, SAVING_FAILED, TRANSACTION_NO T_FOUND, ACCOUNT_NOT_FOUND, LOW_BALANCE, BLOC KED_ACCOUNT ; EN_serverError_t{  typedef enum EN_accountState_t } , RUNNING BLOCKED ; EN_accountState_t{  typedef struct ST_accountsDB_t } ; float balance ; EN_accountState_t state ; uint8_t primaryAccountNumber[20] ; ST_accountsDB_t{  :Use the following prototypes as is .2  EN_transState_t recieveTransactionData(ST_tra ; nsaction_t *transData) EN_serverError_t isValidAccount(ST_cardData_t ; *cardData, ST_accountsDB_t *accountRefrence) EN_serverError_t isBlockedAccount(ST_accounts ; DB_t *accountRefrence) EN_serverError_t isAmountAvailable(ST_termina lData_t *termData, ST_accountsDB_t *accountRe ; frence) EN_serverError_t saveTransaction(ST_transacti ; on_t *transData) ; void listSavedTransactions(void) </pre>

المعايير	يفي بالموصفات
	.You should deliver a screenshot for your server.h file .3
<p>Implement server-side accounts' database</p> <p>.1 Create a global array of <code>ST_accountsDB_t</code> for the valid .accounts database</p> <p><code>;ST_accountsDB_t accountsDB[255]</code></p> <p>.2 .Fill in the array initially with any <b>valid data</b></p> <p>.3 This array has a <b>maximum of 255</b> element/account .data</p> <p>.4 You can fill <b>up to 10 different accounts for the sake of testing</b></p> <p>.5 Example of a <b>running</b> account:</p> <p><code>{2000.0, RUNNING, "8989374615436851"}</code></p> <p>.6 Example of a <b>blocked</b> account, <b>its card is stolen</b>:</p> <p><code>{100000.0, BLOCKED, "5807007076043875"}</code></p> <p>.7 You should deliver a screenshot of your accounts database array with a minimum of at least 5 different accounts for the different test cases, check all needed .test cases in the <b>"Testing the application"</b> section</p>	
<p>Implement server-side transactions' database</p> <p>.1 .Create a global array of <code>ST_transaction_t</code></p> <p>.2 .Fill in the array <b>initially with Zeros</b></p> <p>.3 This array has a <b>maximum of 255</b> .element/transaction data</p> <p>.4 You should deliver a screenshot of your transaction database array</p>	
<p>Implement <code>recieveTransactionData</code> function</p> <p>.1 This function will <b>take all transaction data and</b> .<b>validate its data</b>, it contains all server logic</p> <p>.2 .It checks the account details and amount availability</p> <p>.3 If the account <b>does not exist</b> return <code>FRAUD_CARD</code>, if the <b>amount is not available</b> will return <code>DECLINED_INSUFFECIENT_FUND</code>, if the account is <b>blocked</b> will return <code>DECLINED_STOLEN_CARD</code>, if a</p>	



المعايير	يفي بالموصفات
	<p>transaction can't be saved will return <code>INTERNAL_SERVER_ERROR</code> , else returns <code>APPROVED</code> .</p> <p>.It will update the database with the new balance .4</p> <p>:Test your function .5</p> <p>Create a test function ◦</p> <pre>void recieveTransactionDataTest(void);</pre> <p>to test all possible scenarios, happy-case, .and worst-case scenarios</p> <p>Print all results of your test cases on the ◦ console window, and use the following as :a guide</p> <pre> Tester Name: your name Function Name: recieveTransactionD           ata           :Test Case 1           :Input Data           :Expected Result           :Actual Result           :Test Case 2           :Input Data           :Expected Result           :Actual Result           .           .           .           :Test Case n           :Input Data           :Expected Result           :Actual Result </pre> <p>You should deliver the test function as well as a .6 .screenshot of the results on the console</p>
Implement isValidAccount function	<p>This function will take card data and validate if the .1 .account related to this card exists or not</p>

المعايير	يفي بالموصفات
	<p>2. It checks if the <b>PAN exists or not</b> in the server's .database (<b>searches for the card PAN in the DB</b>)</p> <p>3. If the PAN doesn't exist will return <b>ACCOUNT_NOT_FOUND</b> and the account reference will be <b>NULL</b> , else will return <b>SERVER_OK</b> and return a <b>reference to this account in the DB</b></p> <p>4. Test your function</p> <p>○ Create a test function</p> <p><code>void isValidAccountTest(void);</code> to test all possible scenarios, happy-case, and .worst-case scenarios</p> <p>○ Print all results of your test cases on the console window, and use the following as a guide</p> <pre> Tester Name: your name Function Name: isValidAccount :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n :Input Data :Expected Result :Actual Result </pre> <p>5. You should deliver the test function as well as a .screenshot of the results on the console</p>

المعايير	يفي بالمواصفات
<p>Implement isBlockedAccount function</p>	<p>1. This function <b>takes a reference to the account into</b>  <b>.the database and verifies if it is blocked or not</b>  2. If the account <b>is running</b> it will return <code>SERVER_OK</code>,  else if the account <b>is blocked</b> it will return  <code>BLOCKED_ACCOUNT</code>  3. <b>:Test your function</b></p> <p>Create a test function ◦  <code>void isBlockedAccountTest(void);</code> to  test all possible scenarios, happy-case,  and worst-case scenarios  Print all results of your test cases on the ◦  console window, and use the following as  :a guide</p> <pre> Tester Name: your name Function Name: isBlockedAccount :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n :Input Data :Expected Result :Actual Result </pre> <p>4. You should deliver the test function as well as a  .screenshot of the results on the console</p>

المعايير	يفي بالموصفات
<p>Implement isAmountAvailable function</p>	<p>1. This function will <b>take terminal data and a reference</b> to the account in the database and <b>check if the account has a sufficient amount to withdraw or not</b>.</p> <p>2. It checks if the transaction's amount is available or not.</p> <p>3. If the transaction amount <b>is greater than the balance</b> in the database will return <code>LOW_BALANCE</code>, else will return <code>SERVER_OK</code>.</p> <p>4. Test your function.</p> <p>○ Create a test function</p> <pre>void isAmountAvailableTest(void);</pre> <p>to test all possible scenarios, happy-case, and worst-case scenarios</p> <p>○ Print all results of your test cases on the console window, and use the following as a guide:</p> <pre> Tester Name: your name Function Name: isAmountAvailable :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n :Input Data :Expected Result :Actual Result </pre> <p>5. You should deliver the test function as well as a screenshot of the results on the console.</p>

المعايير	يفي بالموصفات
<p>This function will <b>store all transaction data</b> in the .1 .transactions database</p> <p>It gives a <b>sequence number to a transaction, this .2</b> <b>number is incremented once a transaction is</b> <b>processed</b> into the server, you must check the last sequence number in the server to give the new .transaction a <b>new sequence number</b></p> <p>It saves any type of transaction, <code>APPROVED</code> , .3 <code>DECLINED_INSUFFICIENT_FUND</code> , <code>DECLINED_STOLEN_CARD</code> , <code>FRAUD_CARD</code> , <code>INTERNAL_SERVER_ERROR</code></p> <p>It will list all saved transactions using the .4 <code>listSavedTransactions</code> function</p> <p>Assuming that the connection between the terminal .5 and server is always connected, then it will return <code>SERVER_OK</code></p> <p>.6:Test your function</p> <p>Create a test function ◦</p> <p><code>void saveTransactionTest(void);</code> to test all possible scenarios, happy-case, and .worst-case scenarios</p> <p>Print all results of your test cases on the ◦ console window, and use the following as :a guide</p> <pre> Tester Name: your name Function Name: saveTransaction :Test Case 1 :Input Data :Expected Result :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n </pre>	<p>Implement saveTransaction function</p>

المعايير	يفي بالموصفات
<div data-bbox="302 218 846 375"> :Input Data  :Expected Result  :Actual Result </div> <p data-bbox="292 417 938 495">You should deliver the test function as well as a .7 .screenshot of the results on the console</p>	
<p data-bbox="280 634 935 800">This function <b>prints</b> all transactions found in the .1 .transactions DB Please follow the following format for only one .2 :transaction data</p> <div data-bbox="207 810 907 1295"> <pre>##### :Transaction Sequence Number :Transaction Date :Transaction Amount :Transaction State :Terminal Max Amount :Cardholder Name :PAN :Card Expiration Date #####</pre> </div> <p data-bbox="662 1323 938 1356">:Test your function .3</p> <p data-bbox="566 1404 883 1436">Create a test function ◦</p> <div data-bbox="318 1449 846 1482"> <pre>void listSavedTransactionsTest(void);</pre> </div> <p data-bbox="311 1495 846 1568">to test all possible scenarios, happy-case, .and worst-case scenarios</p> <p data-bbox="311 1581 883 1703">Print all results of your test cases on the ◦ console window, and use the following as :a guide</p> <div data-bbox="302 1713 846 2003"> <pre>Tester Name: your name Function Name: listSavedTransaction ns :Test Case 1 :Input Data :Expected Result</pre> </div>	<p data-bbox="1052 634 1336 753">Implement listSavedTransactions function</p>

المعايير	يفي بالمواصفات
<pre> :Actual Result :Test Case 2 :Input Data :Expected Result :Actual Result . . . :Test Case n :Input Data :Expected Result :Actual Result </pre> <p>You should deliver the test function as well as a .4 .screenshot of the results on the console</p>	
<p>Record a video where you discuss each function you .1 .implemented in this module .2.Explain and execute all test functions you made .3.The video is <b>4 minutes maximum</b> .4.You may record it in Arabic or English .5.Muted videos will not be acceptable .6. <b>All of the above are mandatory to pass this</b> .criterion</p>	<p>Explain your work</p>

### Implement the application

المعايير	يفي بالمواصفات
<p>1. Use the following prototypes as-is</p> <pre>void appStart(void);</pre>	<p>Fill in application.h file with functions' prototypes</p>

المعايير	يفي بالمواصفات
	You should deliver a screenshot for your application.h2 .file
Implement appStart function	Please refer to the <b>flow chart</b> attached under the .1 <b>instructions video</b> in order to implement this .application .2 You should deliver <b>all project folders and files</b>
Explain your work	Record a video where you discuss each function you .1 .implemented in this module .2 The video is <b>2 minutes maximum</b> .3 You may record it in Arabic or English .4 Muted videos will not be acceptable .5 All of the above are mandatory to pass this criterion

## Testing the application

المعايير	يفي بالمواصفات
Transaction approved user story	As a bank customer have an account and has a <b>valid</b> .1 and <b>not expired card</b> , I want to withdraw an <b>amount</b> of money <b>less than the maximum allowed</b> and <b>less</b> <b>than or equal to the amount in my balance</b> , so that I am expecting that the transaction is <b>approved</b> and my <b>account balance is reduced by the withdrawn</b> <b>.amount</b> .2 You should deliver a screenshot of the test result on .the console
Exceed the maximum	As a bank customer have an account, that has a <b>valid</b> .1 and <b>not expired card</b> , I want to withdraw an amount of



المعايير	يفي بالمواصفات
amount user story	money that <b>exceeds the maximum allowed amount</b> .so that I am expecting the transaction <b>declined</b> You should deliver a screenshot of the test result on .2 .the console
Insufficient fund user story	As a bank customer have an account and has a <b>valid .1 and not expired card</b> , I want to withdraw an amount of money <b>less than the maximum allowed and larger than the amount in my balance</b> so that I am expecting .that the transaction <b>declined</b> You should deliver a screenshot of the test result on .2 .the console
Expired card user story	As a bank customer have an account and a <b>valid but .1 expired card</b> , I want to withdraw an amount of money .so that I expect that the transaction <b>declined</b> You should deliver a screenshot of the test result on .2 .the console
Stolen card user story	As a bank customer have an account and has a <b>valid .1 and not expired but stolen card</b> , I want to block anyone from using my card so that I am expecting that .any transaction made by this card is <b>declined</b> You should deliver a screenshot of the test result on .2 .the console
Explain your work	.Record a video where you discuss each test case .1 .The video is <b>4 minutes maximum</b> .2 .You may record it in Arabic or English .3 .Muted videos will not be acceptable .4 .All of the above are mandatory to pass this criterion .5

## اقتراحات لجعل مشروعك متميزًا!

1. In getCardPAN function
    - Provide the PAN as a Luhn number
  2. Implement all optional functions
  3. For the server-side accounts DB
    - Instead of a global array create a text file "Accounts DB.txt" that stores all account data and read this file into your application
  4. For server-side transactions DB
    - Instead of a global array create a text file "Transactions DB.txt" where you will save all transactions and read if you need
  5. Test your application against the Fraud card
- As a bank administrator, I want to issue my own cards, so I am expecting that
- any transaction made by any fraud card (failed in Luhn check) is declined