|  |  |  |  |
| --- | --- | --- | --- |
| **المملكة العربية السعودية السعودية**  **وزارة التعليم العالي**  **جامعة الإمام محمد بن سعود الإسلامية**  **كلية علوم الحاسب والمعلومات** | | A description...  **Second term 1441/2020** | **KINGDOM OF SAUDI ARABIA**  **Ministry of Higher Education**  **Al-Imam Mohammad University**  **College of Computer & Information Sciences** |
|  | **Software Engineering (CS- 310)**  **Section: 171**  **Project-Phase No: 2**  **Computer Recommendation Software (CRS)**  **(Design Document)**  **Submitted By**  Saad BinOnayq (439017145) – Coordinator  Fawzan alhantoshi (439014363)  Mohammed Alkhalifah (439011298)  Tareq Alagel (433000438)  Khaled Almodameeg (438013580)  **Supervisor**  Sultan Alqahtanie  **Date:** 2020\2\10 | | | |

**Table of Contents**

[**Abstract**](#_ql12tnvxzeic) **3**

[**1. Introduction**](#_ycua9umx8iuy) **4**

[1.1 Purpose](#_6hkgew9mxwqq) **5**

[1.2 Scope](#_s5uz8tliod72) **5**

[1.3 Definitions, Acronyms, and Abbreviations](#_cgou6ijju4t2) **5**

[1.4 References](#_4oqw3z8bd6yu) **5**

[1.5 Overview](#_lhht5u3qosok) **5**

[**2. Design**](#_82p6xwt2bnqu) **5**

[2.1 System Level Diagram](#_yxtrdozfrype) **6**

[2.1.1 Use-Case Diagram](#_f7mbrcbwzyi0) **6**

[2.1.2 MVC Pattern](#_z0ma95hhq86b) **6**

[2.2 Class Diagram](#_if2p0kr0zcbq) **7**

[2.3 Class Method Description](#_lbuvuinrf0vs) **8**

[2.3.1 ProductListAdapter](#_sokouq4v2107) **8**

[2.3.2 ResultActivity](#_favsqobjgcz2) **9**

[2.3.3 MainActivity](#_hu9en6p4j28b) **10**

[2.4 State Charts](#_nrav5onr7gz1) **21**

[2.5 Flowchart Diagram](#_1g4ytyl2ho26) **22**

[2.6 User Interface Design](#_h5swkli4kmj1) **23**

[**3. Contributions of team members**](#_c0a8xkcfhb8c) **27**

[**4. Conclusion**](#_mwsuistxcm4n) **27**

****

## 

## **Abstract**

*This design document shows Computer recommendation software (CRS)**in an architectural approach, the document will graphically describe the classes, methods will be described in a table view.*

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **1. Introduction**

*In phase one (SRS) we describe the CRS application and how it will serve the end user, in DD we will define the System architecture and models in a graphical way.*

## **1.1 Purpose**

*The purpose of DD is to describe the design of “Computer recommendation software” and its inputs and outputs for the first version in detail.*

## **1.2 Scope**

*The “Computer Recommendation Software” is a mobile application which helps people to find computers that fit their field of interests. By recommending a system based on the user’s specification, and combine all components in a single convenient page. It also suggests some applications to the user based on the type of the recommended system. The application should also be free and downloadable through a mobile phone application store or similar services.*

*All items will be provided from websites (such as Newegg*[*[3]*](#_g0eicvtyem5x)*) through API. Then, this application will simply process through all the data and display them to the user.*

*Furthermore, this software requires internet connection to be able to connect to websites to get all the needed data for this application. It also has the capability to get all items by their best value based on the user's region. And for each item it also includes a brief description, summary and pros and cons of the recommended system.*

## **1.3 Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definitions** |
| CRS | *Computer Requirement Software “This Application”* |
| SRS | *System Requirement Specification* |
| *MySQL* | *An open source database management system (RDBMS)* |
| *DD* | *Design document* |
| *URL* | Uniform Resource Locator, colloquially termed a web address,is a reference to a web resource that specifies its location on computer network |

## **1.4 References**

*[1] Android developers, Founded in 2004,* [*https://developer.android.com/*](https://developer.android.com/)

## **1.5 Overview**

*In section two we will describe graphically the System level diagram, Class diagram, Class method description (Table view), State charts, Flowchart diagram and User interface design. The third section provides contributions of team members and the conclusion.*

## 

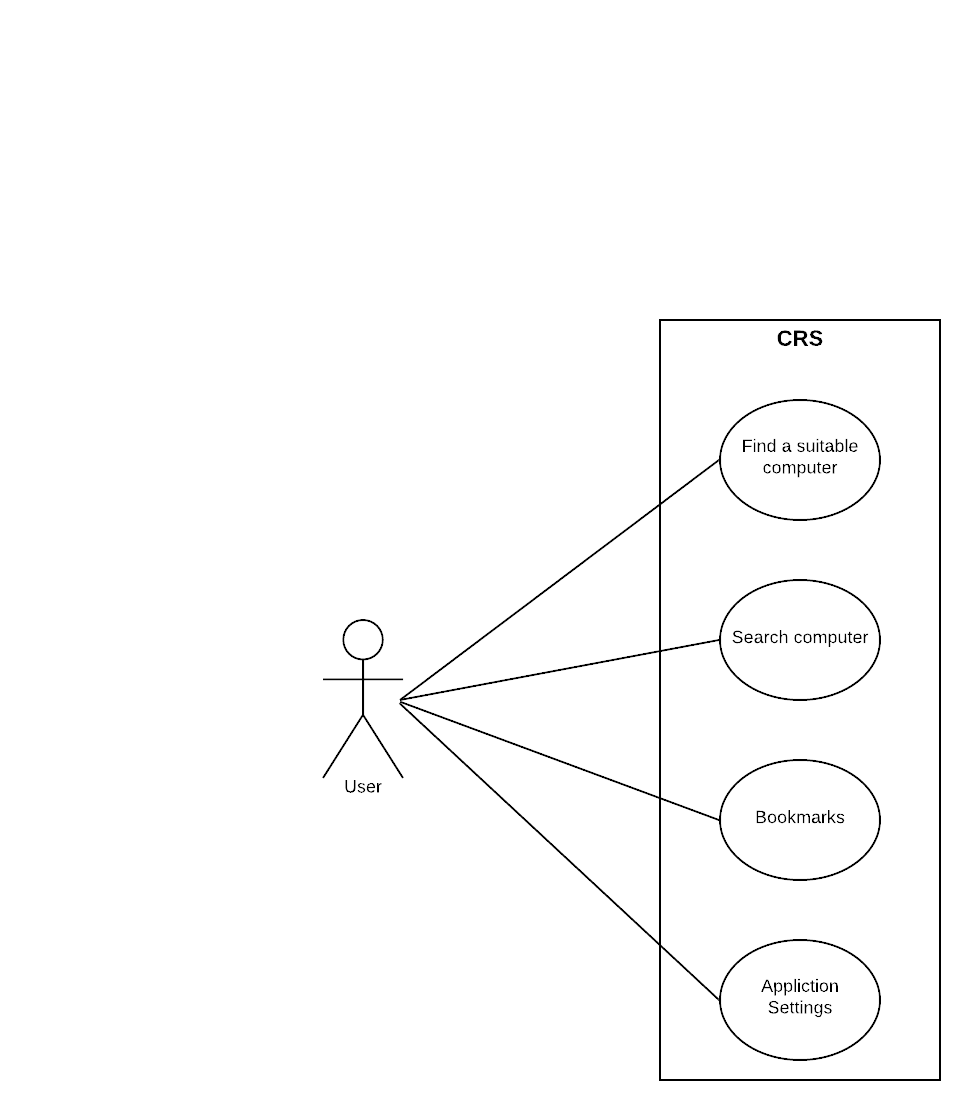
## **2. Design**

*This section will graphically describe all the designs in DD.*

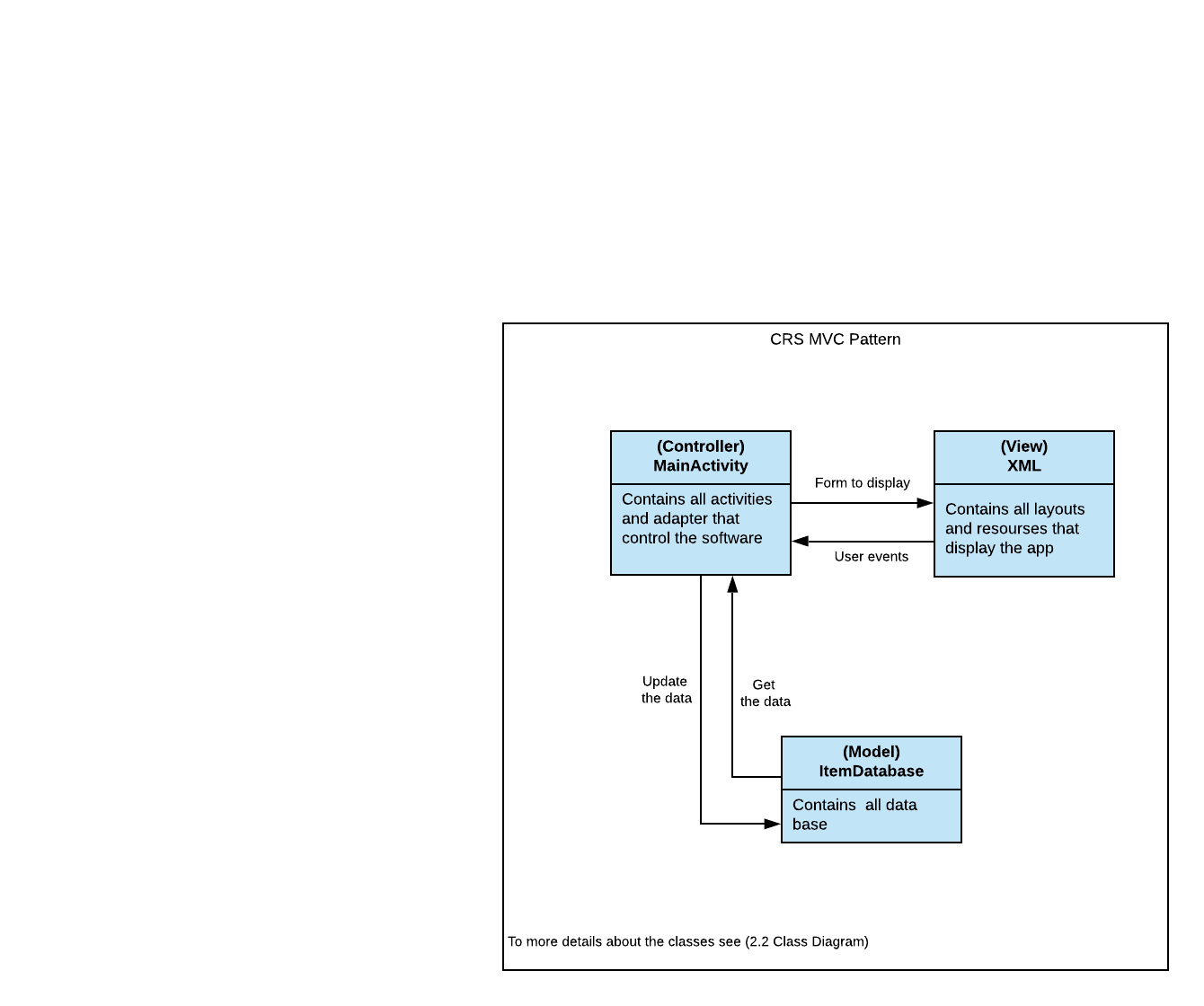
## **2.1 System Level Diagram**

*This subsection will be considered in the System level diagram (Use Case and MVC pattern).*

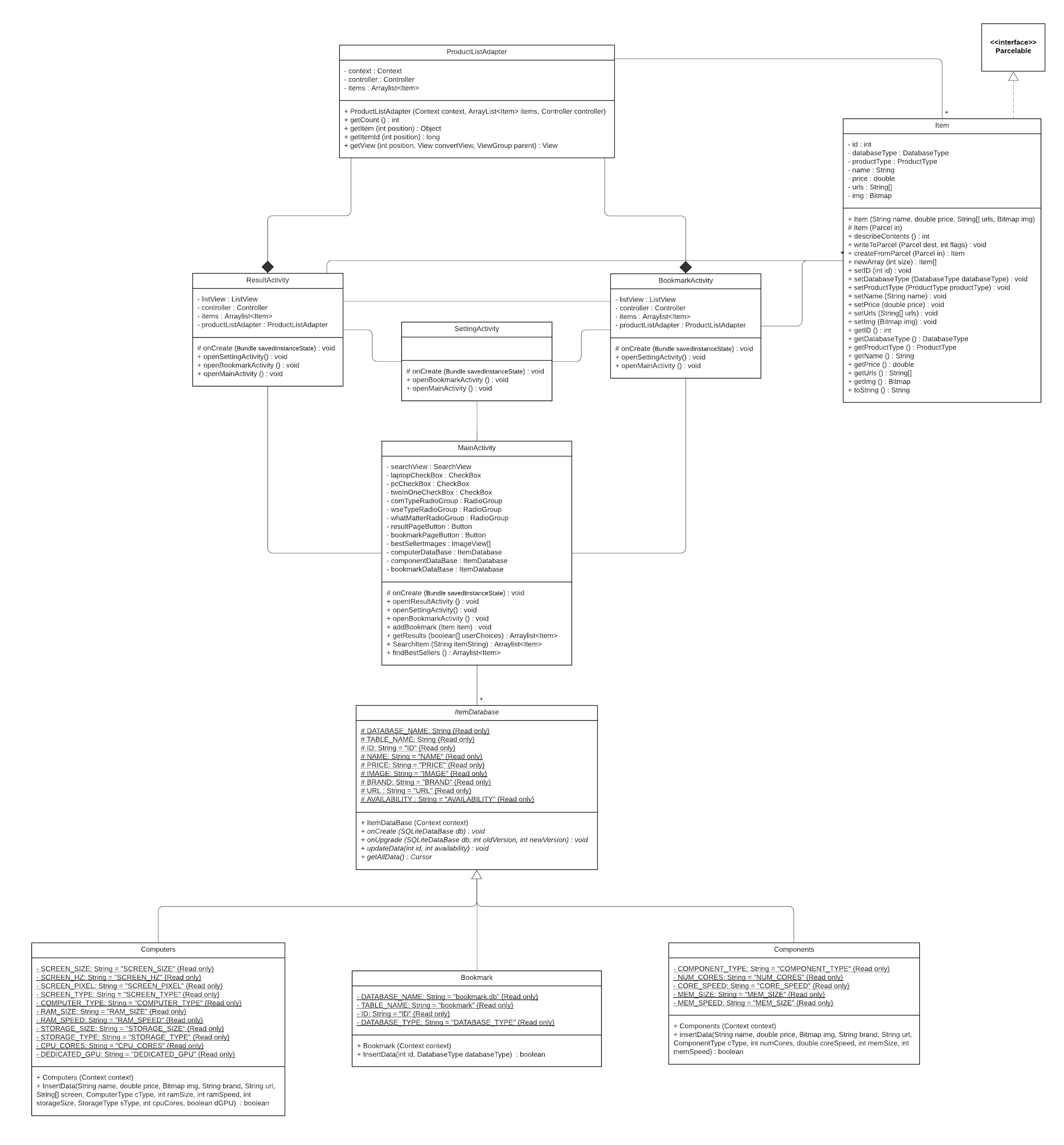
## ***2.1.1 Use-Case Diagram***



## 2.1.2 MVC Pattern

******

## **2.2 Class Diagram**

[](https://drive.google.com/open?id=1f0GsYjIWWz3ULO8M2Fbcy-AKHqDLf151)

**(Click the image for higher quality)**

## **2.3 Class Method Description**

*This section to list all the methods of the classes.*

## ***2.3.1 ProductListAdapter***

*This class to customize the view of list adapter.*

|  |  |
| --- | --- |
| **Class** | ProductListAdapter |
| **Method** | ProductListAdapter |
| **Visibility** | Public |
| **Return Type** | Constructor |
| **Parameters types** | Context:context, ArrayList:items, Controller:controller |
| **Description** | This is the Constructor for the ProductListAdapter class. |

|  |  |
| --- | --- |
| **Class** | ProductListAdapter |
| **Method** | getCount |
| **Visibility** | Public |
| **Return Type** | int |
| **Parameters types** | None |
| **Description** | get the size of the ArrayList items. |

|  |  |
| --- | --- |
| **Class** | ProductListAdapter |
| **Method** | getItem |
| **Visibility** | Public |
| **Return Type** | Object |
| **Parameters types** | int:position |
| **Description** | get the item by its position in the ArrayList items. |

|  |  |
| --- | --- |
| **Class** | ProductListAdapter |
| **Method** | getItemId |
| **Visibility** | Public |
| **Return Type** | long |
| **Parameters types** | int:position |
| **Description** | get the item Id by its position in the ArrayList items. |

|  |  |
| --- | --- |
| **Class** | ProductListAdapter |
| **Method** | getView |
| **Visibility** | Public |
| **Return Type** | View |
| **Parameters types** | int:position, View:convertView, ViewGroup:parent |
| **Description** | Get a view that displays the data at the specified position in the data set. |

## ***2.3.2 ResultActivity***

*This class to view the result page.*

|  |  |
| --- | --- |
| **Class** | ResultActivity |
| **Method** | onCreate |
| **Visibility** | Protected |
| **Return Type** | void |
| **Parameters types** | Bundle:savedInstanceState |
| **Description** | This method to initialize the class. |

|  |  |
| --- | --- |
| **Class** | ResultActivity |
| **Method** | openSettingActivity |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | None |
| **Description** | To open the settings of the result activity. |

|  |  |
| --- | --- |
| **Class** | ResultActivity |
| **Method** | openBookmarkActivity |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | None |
| **Description** | To open the bookmark of the result activity. |

|  |  |
| --- | --- |
| **Class** | ResultActivity |
| **Method** | openMainActivity |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | None |
| **Description** | To open the main of the result activity. |

## ***2.3.3 MainActivity***

*This class to view the main page.*

|  |  |
| --- | --- |
| **Class** | MainActivity |
| **Method** | onCreate |
| **Visibility** | Protected |
| **Return Type** | void |
| **Parameters types** | Bundle:savedInstanceState |
| **Description** | This method to initialize the class. |

|  |  |
| --- | --- |
| **Class** | MainActivity |
| **Method** | opentResultActivity |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | None |
| **Description** | To open the result activity class. |

|  |  |
| --- | --- |
| **Class** | MainActivity |
| **Method** | openSettingsActivity |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | None |
| **Description** | To open the settings activity class. |

|  |  |
| --- | --- |
| **Class** | MainActivity |
| **Method** | openBookmarkActivity |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | None |
| **Description** | To open the bookmark activity class. |

|  |  |
| --- | --- |
| **Class** | MainActivity |
| **Method** | addBookmark |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | Item:Item |
| **Description** | This method adds a bookmark. |

|  |  |
| --- | --- |
| **Class** | MainActivity |
| **Method** | getResults |
| **Visibility** | Public |
| **Return Type** | Arraylist<Item> |
| **Parameters types** | boolean[] userChoices |
| **Description** | This method gets the result depending on the user choices. |

|  |  |
| --- | --- |
| **Class** | MainActivity |
| **Method** | SearchItem |
| **Visibility** | Public |
| **Return Type** | Arraylist<Item> |
| **Parameters types** | String itemString |
| **Description** | This method gets the result depending on the user choices. |

|  |  |
| --- | --- |
| **Class** | MainActivity |
| **Method** | findBestSellers |
| **Visibility** | Public |
| **Return Type** | Arraylist<Item> |
| **Parameters types** | None |
| **Description** | This method to find the best sellers. |

***2.3.4 BookmarkActivity***

*This class to view the Bookmark page.*

|  |  |
| --- | --- |
| **Class** | BookmarkActivity |
| **Method** | onCreate |
| **Visibility** | Prptected |
| **Return Type** | void |
| **Parameters types** | Bundle:savedInstanceState |
| **Description** | This method to initialize the class. |

|  |  |
| --- | --- |
| **Class** | BookmarkActivity |
| **Method** | openSettingsActivity |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | None |
| **Description** | To open the settings activity class. |

|  |  |
| --- | --- |
| **Class** | BookmarkActivity |
| **Method** | openMainActivity |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | None |
| **Description** | To open the main activity class. |

***2.3.5 SettingActivity***

*This class to view the settings page.*

|  |  |
| --- | --- |
| **Class** | SettingActivity |
| **Method** | onCreate |
| **Visibility** | Protected |
| **Return Type** | void |
| **Parameters types** | Bundle:sacedInstanceState |
| **Description** | This method to initialize the class. |

|  |  |
| --- | --- |
| **Class** | SettingActivity |
| **Method** | openBookmarkActivity |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | None |
| **Description** | This method opens the BookmarkActivity class. |

|  |  |
| --- | --- |
| **Class** | SettingActivity |
| **Method** | openMainActivity |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | None |
| **Description** | This method opens the MainActivity class. |

***2.3.6 Item***

*This class temporarily holds an item from the database.*

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | Item |
| **Visibility** | Public |
| **Return Type** | Constructer |
| **Parameters types** | String:name, double:price, String[]:urls, Bitmap:img |
| **Description** | To construct the class with several parameters. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | Item |
| **Visibility** | Protected |
| **Return Type** | Constructer |
| **Parameters types** | Parcel:in |
| **Description** | To construct the class with Parcel. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | describeContents |
| **Visibility** | Public |
| **Return Type** | int |
| **Parameters types** | None. |
| **Description** | Describe the kinds of special objects contained in this parcelable instance’s marshaled representation. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | writeToParcell |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | Parcel:dest, int:flags |
| **Description** | Flatten this object into a parcel. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | createFromParcel |
| **Visibility** | Public |
| **Return Type** | Item |
| **Parameters types** | Parcel:in |
| **Description** | Create an item from Parcel interface. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | setID |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | int:id |
| **Description** | To set the item’s ID. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | setDatabaseType |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | DatabaseType:databaseType |
| **Description** | To set the item’s Database type. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | setProductType |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | ProductType:productType |
| **Description** | To set the item’s product type. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | setName |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | String:name |
| **Description** | To set the item’s name. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | setPrice |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | double price |
| **Description** | To set the item’s price. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | setUrls |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | String[]:urls |
| **Description** | To set the item’s URL. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | setImg |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | Bitmap:img |
| **Description** | To set the item’s image. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | getID |
| **Visibility** | Public |
| **Return Type** | int |
| **Parameters types** | None |
| **Description** | To get the item’s ID. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | getDatabaseType |
| **Visibility** | Public |
| **Return Type** | DatabaseType |
| **Parameters types** | None |
| **Description** | To get the item’s Database type. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | getProductType |
| **Visibility** | Public |
| **Return Type** | ProductType |
| **Parameters types** | None |
| **Description** | To get the item’s product type. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | getName |
| **Visibility** | Public |
| **Return Type** | String |
| **Parameters types** | None |
| **Description** | To get the item’s name. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | getprice |
| **Visibility** | Public |
| **Return Type** | double |
| **Parameters types** | None |
| **Description** | To get the item’s price. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | getUrls |
| **Visibility** | Public |
| **Return Type** | String[] |
| **Parameters types** | None |
| **Description** | To get the item’s URL. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | getImg |
| **Visibility** | Public |
| **Return Type** | Bitmap |
| **Parameters types** | None |
| **Description** | To get the item’s image. |

|  |  |
| --- | --- |
| **Class** | Item |
| **Method** | toString |
| **Visibility** | Public |
| **Return Type** | String |
| **Parameters types** | None |
| **Description** | To print the item’s info. |

***2.3.7 ItemDatabase***

*This class temporarily holds an item from the database.*

|  |  |
| --- | --- |
| **Class** | ItemDatabase |
| **Method** | ItemDataBase |
| **Visibility** | Public |
| **Return Type** | String |
| **Parameters types** | Context:context |
| **Description** | To construct the class. |

|  |  |
| --- | --- |
| **Class** | ItemDatabase |
| **Method** | onCreate |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | SQLiteDataBase:db |
| **Description** | To initial the database. |

|  |  |
| --- | --- |
| **Class** | ItemDatabase |
| **Method** | onUpgrade |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | SQLiteDataBase:db, int:oldVersion, int:newVersion |
| **Description** | To upgrade the database. |

|  |  |
| --- | --- |
| **Class** | ItemDatabase |
| **Method** | getAllData |
| **Visibility** | Public |
| **Return Type** | Cursor |
| **Parameters types** | None |
| **Description** | To get all the data from the database. |

***2.3.8 Bookmark***

*This class manages the bookmark database.*

|  |  |
| --- | --- |
| **Class** | Bookmark |
| **Method** | Bookmark |
| **Visibility** | Public |
| **Return Type** | Constructer |
| **Parameters types** | Context:context |
| **Description** | To construct the bookmark class using context. |

|  |  |
| --- | --- |
| **Class** | Bookmark |
| **Method** | InsertData |
| **Visibility** | Public |
| **Return Type** | boolean |
| **Parameters types** | int:id, DatabaseType:databaseType |
| **Description** | This method insert data into database. |

***2.3.9 Components***

*This class manages the components database.*

|  |  |
| --- | --- |
| **Class** | Components |
| **Method** | Components |
| **Visibility** | Public |
| **Return Type** | Constructer |
| **Parameters types** | Context:context |
| **Description** | To construct the components class using context. |

|  |  |
| --- | --- |
| **Class** | Components |
| **Method** | insertData |
| **Visibility** | Public |
| **Return Type** | void |
| **Parameters types** | String:name, double:price, Bitmap:img, String:brand, String:url, ComponentType:cType, int:numCores, double:coreSpeed, int:memSize, int:memSpeed |
| **Description** | To insert data into the database. |

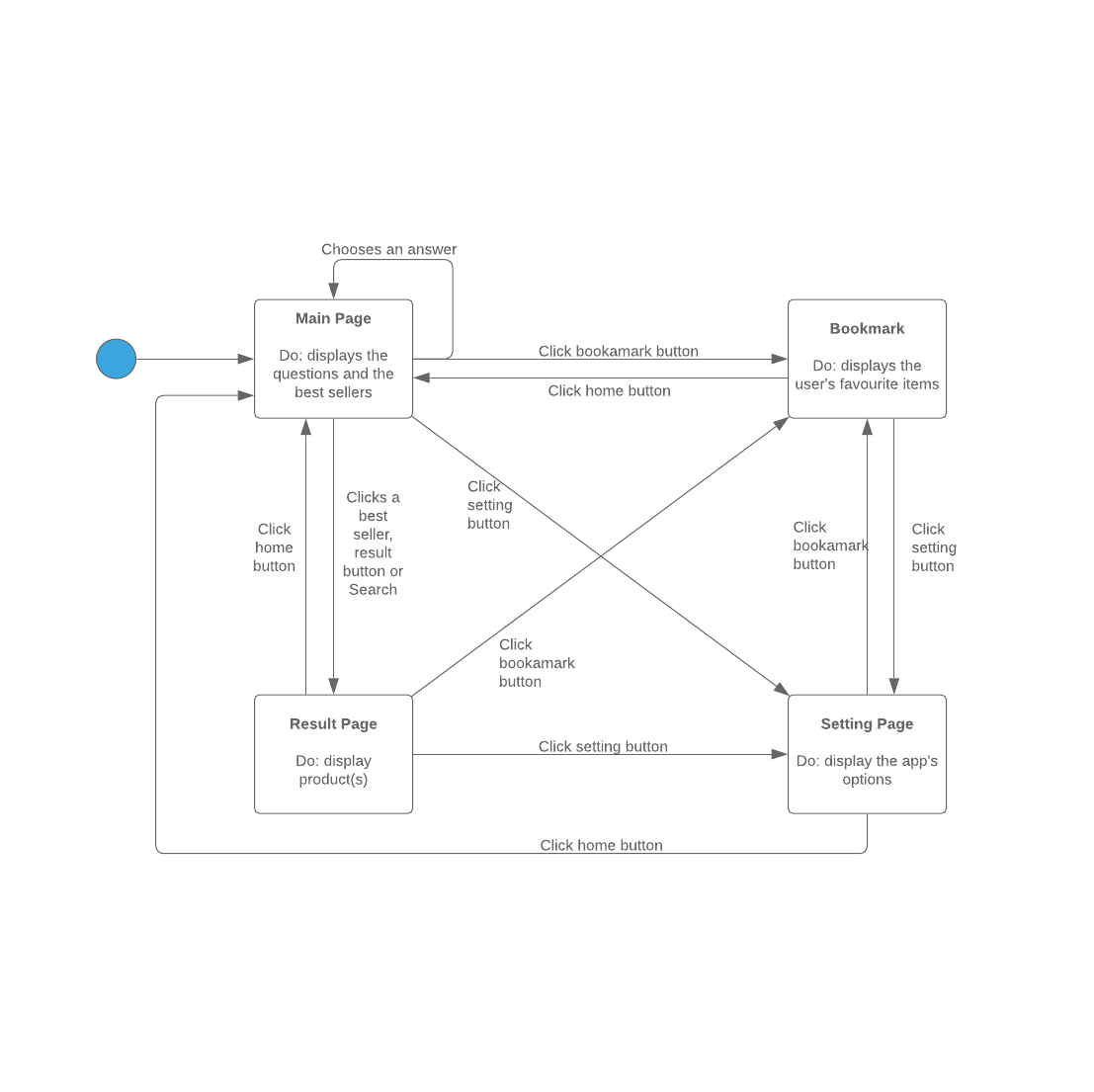
***2.3.10 Computers***

*This class manages the computers database.*

|  |  |
| --- | --- |
| **Class** | Computers |
| **Method** | Computers |
| **Visibility** | Public |
| **Return Type** | Constructer |
| **Parameters types** | Context:context |
| **Description** | To construct the computer class. |

|  |  |
| --- | --- |
| **Class** | Computers |
| **Method** | insertData |
| **Visibility** | Public |
| **Return Type** | boolean |
| **Parameters types** | String: name, double: price, Bitmap: img, String: brand, String:url, String[]: screen, ComputerType: cType, int: ramSize, int:ramSpeed, int: storageSize, StorageType: sType, int: cpuCores, boolean: dGPU |
| **Description** | This method inserts data into the database. |

## **2.4 State Charts**



## **2.5 Flowchart Diagram**

## 

## **2.6 User Interface Design**

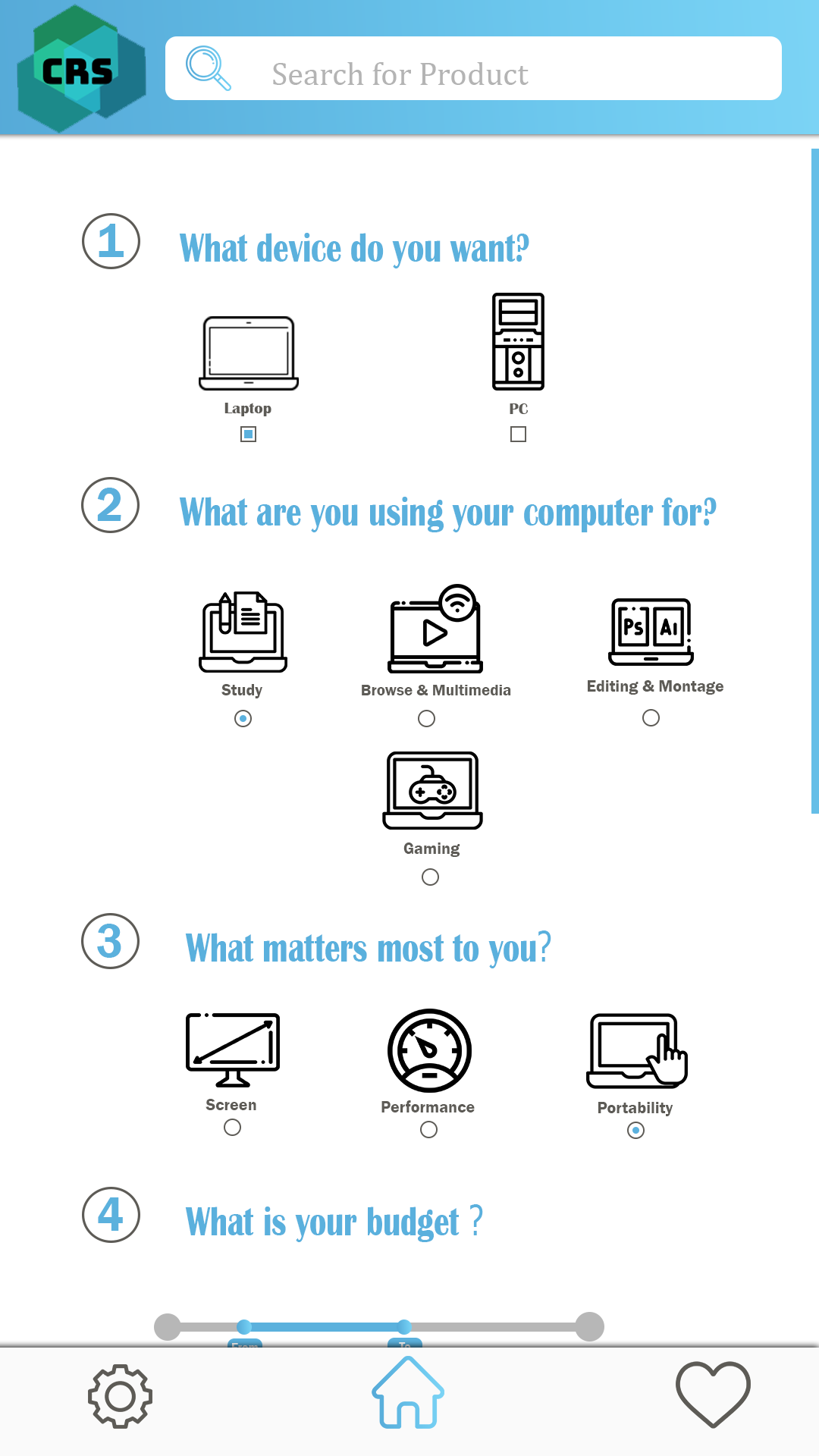


Figure.2

*This is Home page With four Question and Search bar and windows buttons*

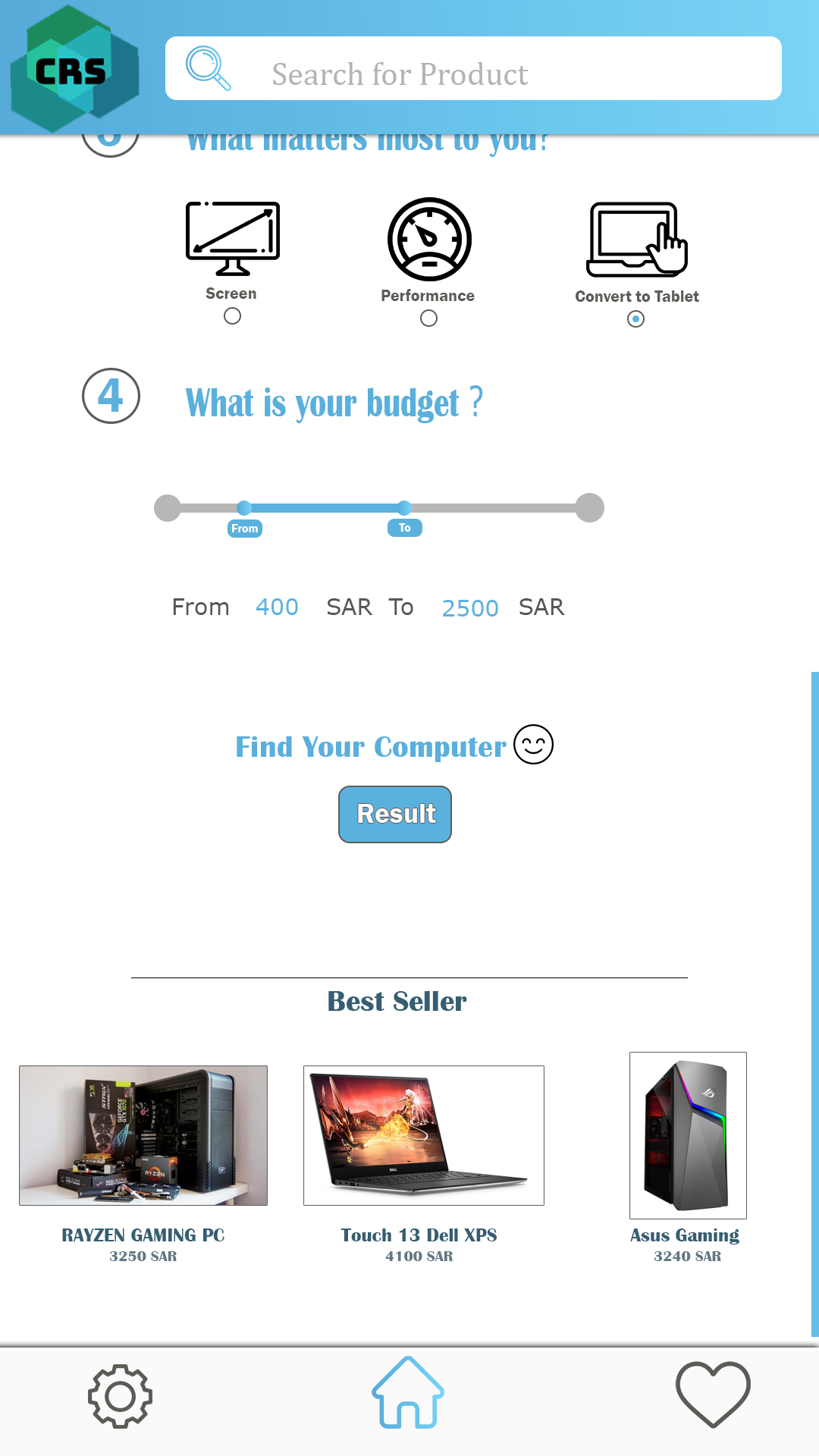
****

Figure.3

*Follow of the Home Page with result button*

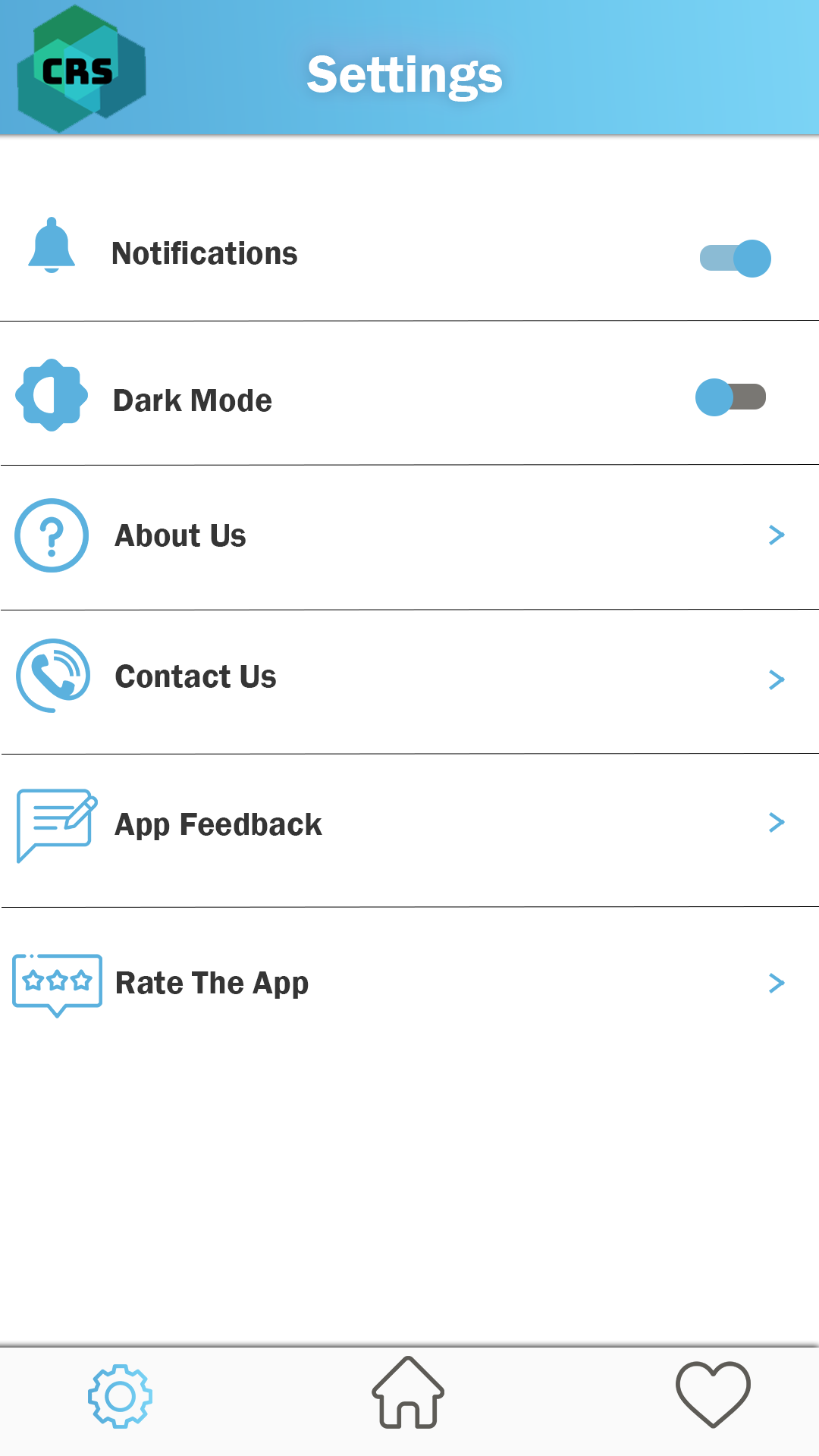


Figure.4

*This is settings window with some options*

*(notifications- dark mode-about us-contact us-app feedback-rate the app)*

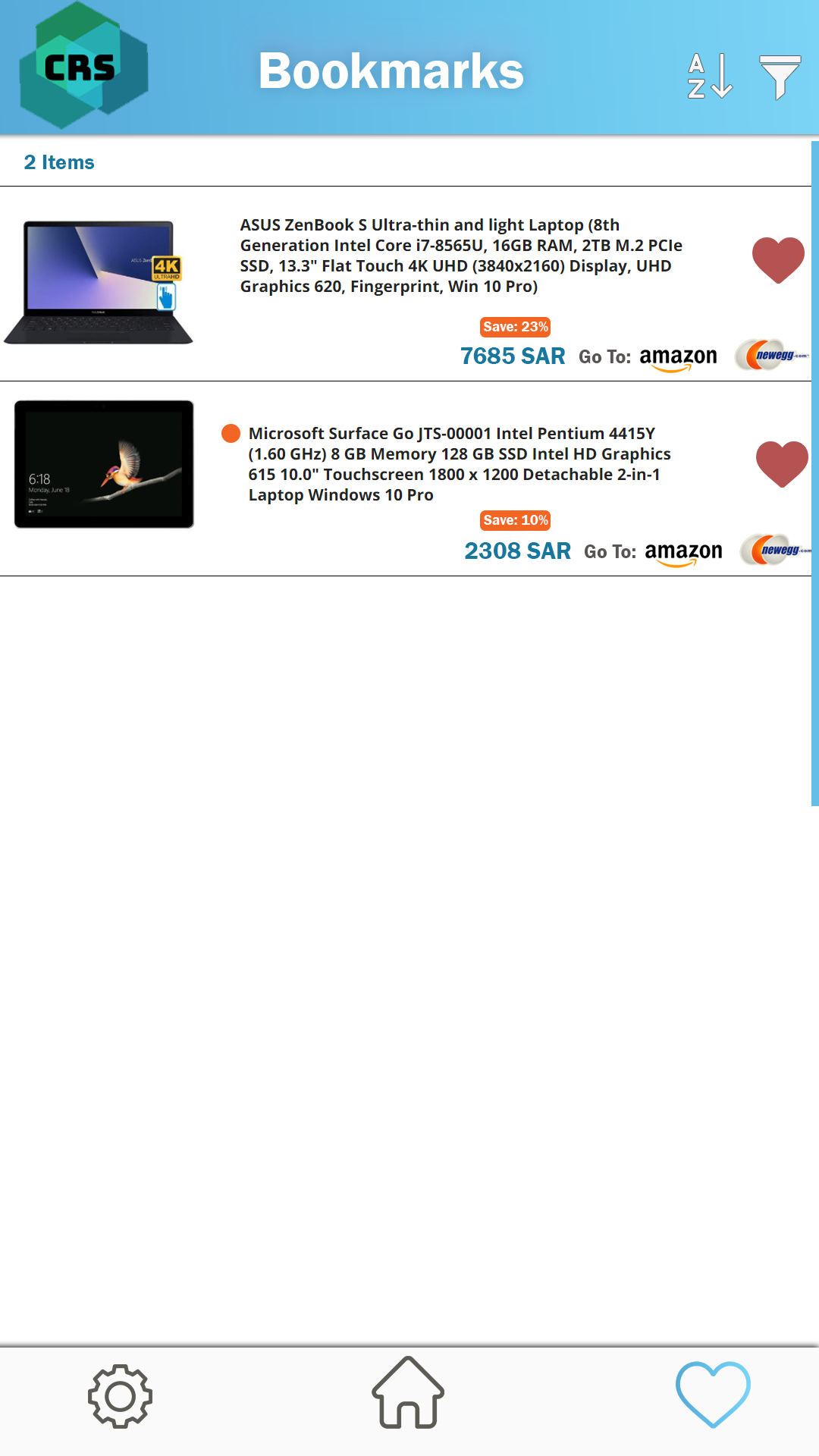


Figure.5

*This is bookmark window with sort and buttons*

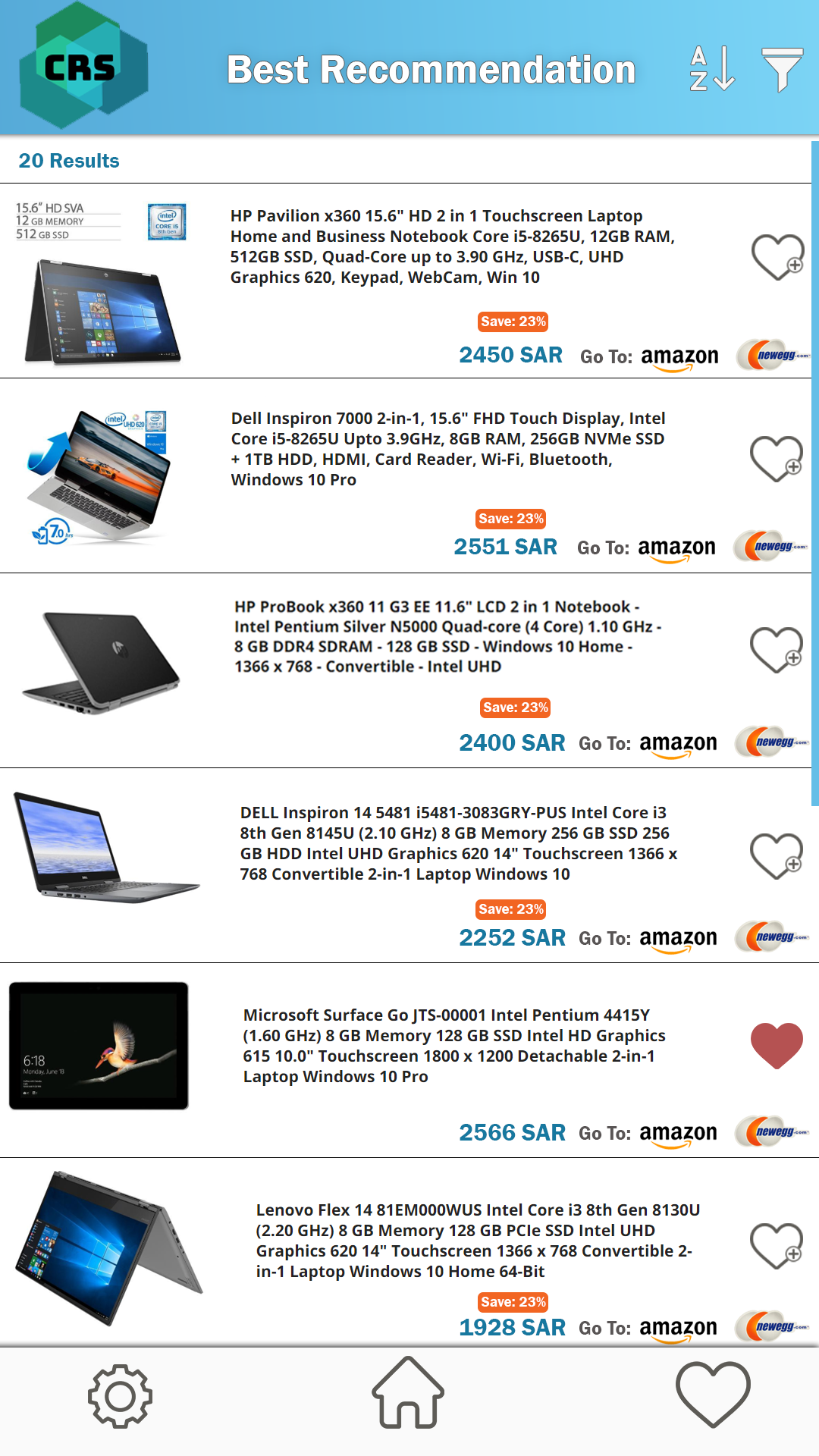
****

Figure.6

*This is the result window from the result button. with sort and buttons*

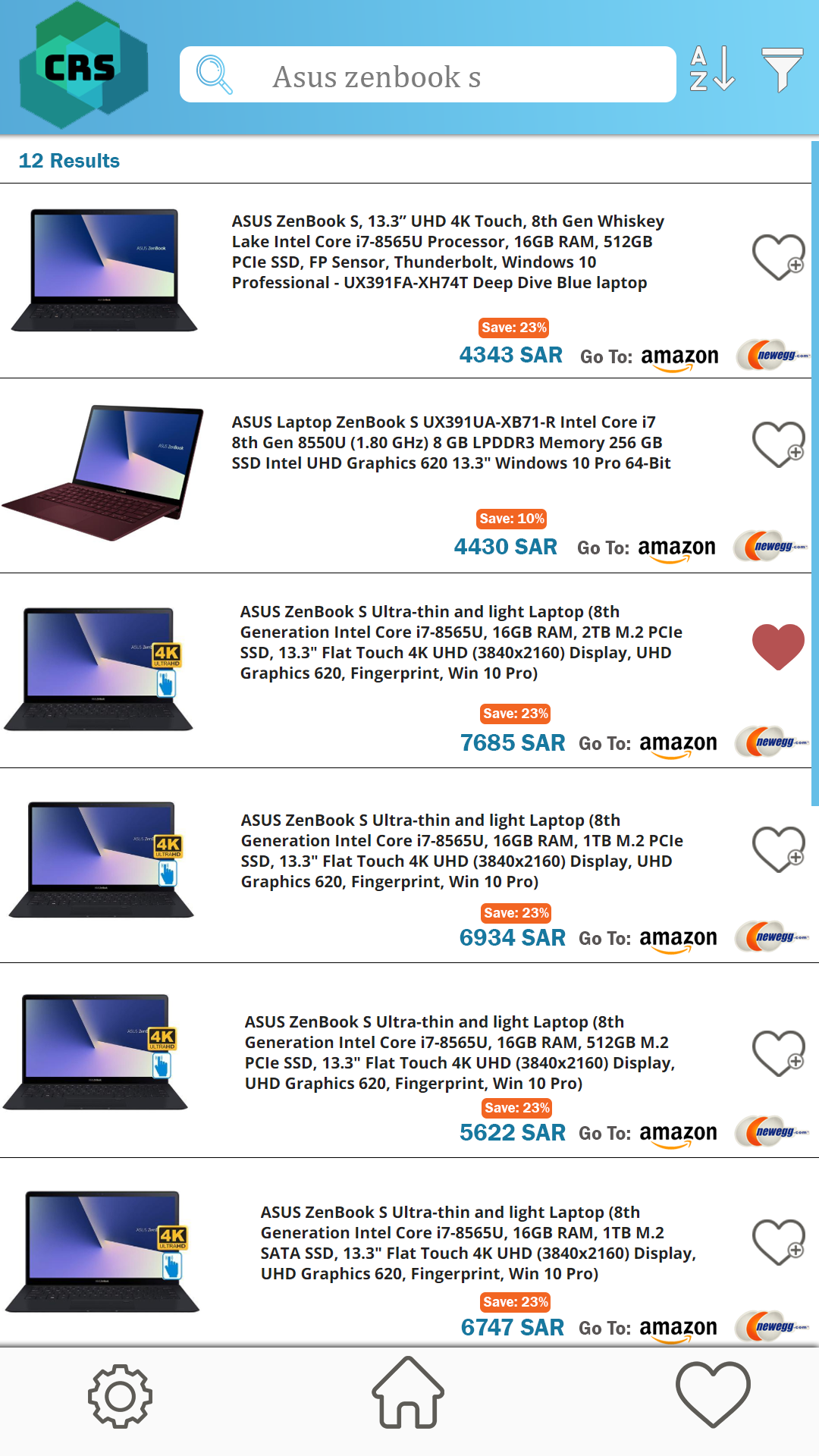


Figure.7

*This is the result window form product search. with sort and buttons*

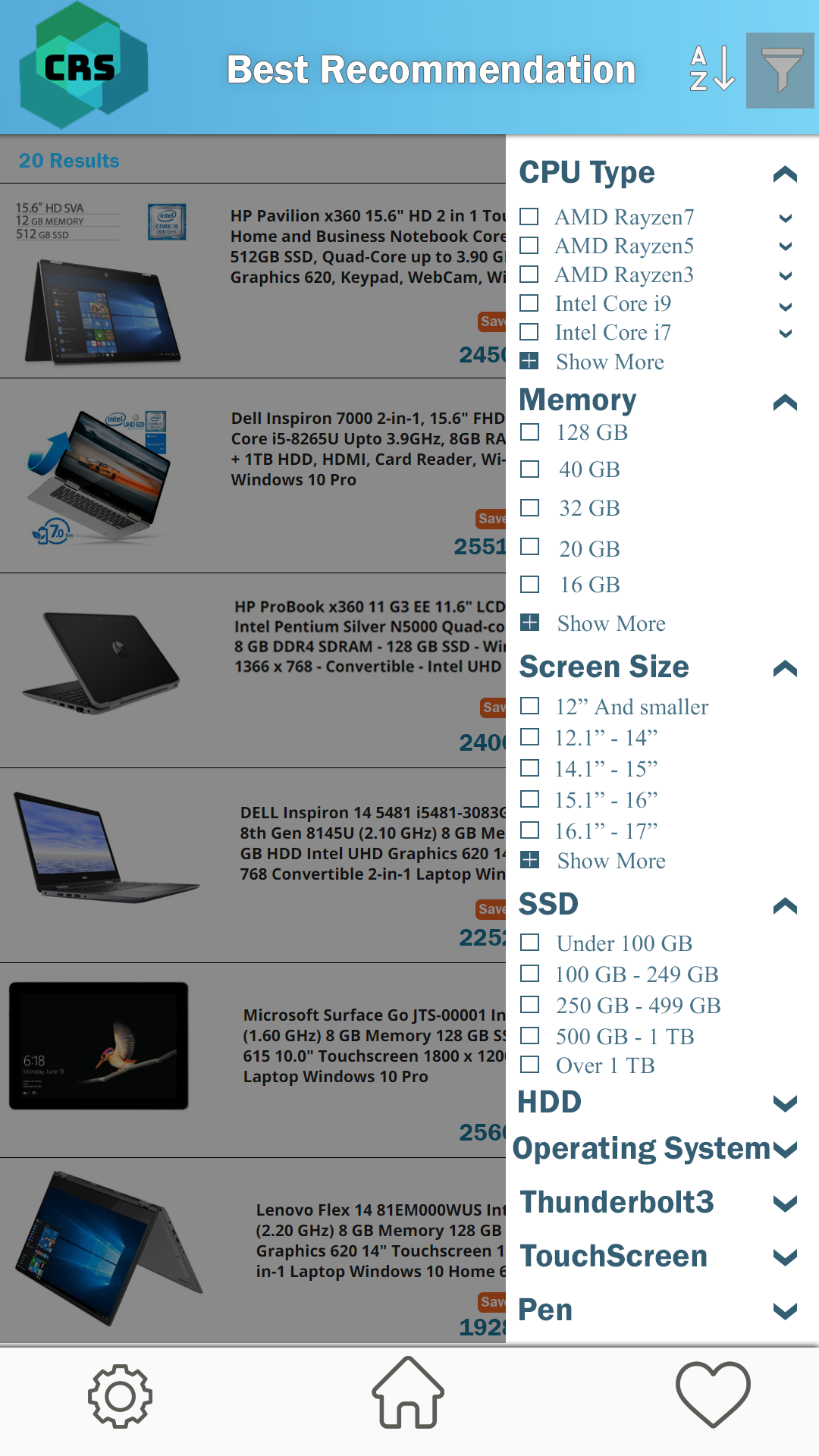


Figure.8

*This is filter option window with a lot of attributes*

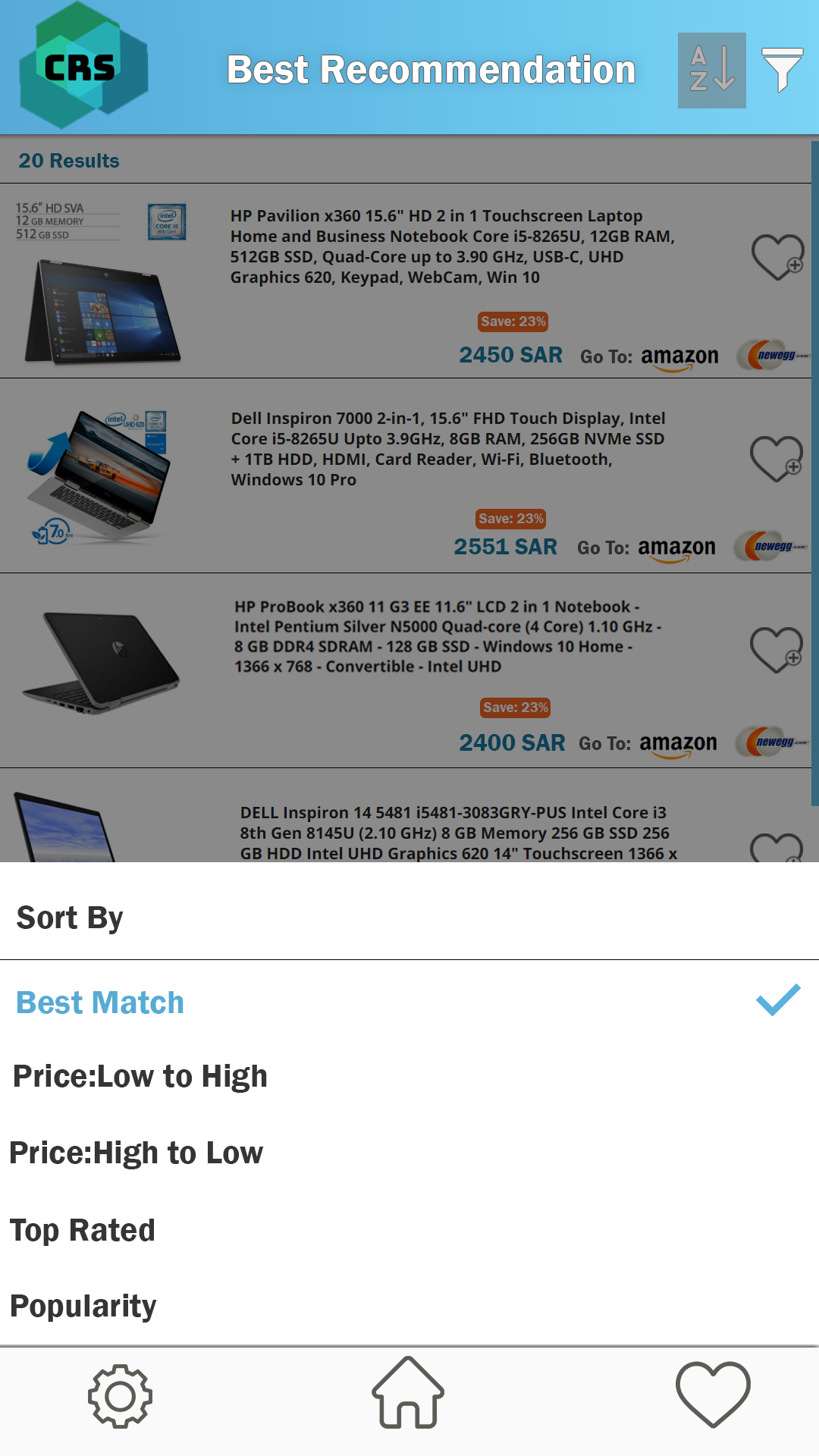


Figure.8

*This is sort option window with five options*

## **3. Contributions of team members**

|  |  |
| --- | --- |
| Saad Bin Onayq | Introduction, Definitions, User interface and System level diagram (Use case and MVC pattern) |
| Fowzan Alhantoshi | Abstract, Introduction, Definitions, References, Overview, Class method description, Contribution of team members and conclusion |
| Mohammad Alkhalifah | Table of content, Purpose, Scope, Definitions, Class diagram, State charts and flowchart diagram |
| Khaled Almodameeg | Class method description |
| Tareq Alagel |  |

**GitHub Repository** : https://github.com/Saad-BinMansour/CRS

## 

## **4. Conclusion**

*The purpose of this DD (Design Document) is to show the system of CRS in an architectural view with different diagrams, modules and show the user interface. Also Explain what each method does in the system.*