**Requirements :**

Functional Requirements - What must it do?

The system must …

* Register users
* enable users to upload/book items
* allow users to select the needed items from the listed items
* allow users to include a description and contact information while uploading an item.
* provide users with the ability to add chosen items to the cart and book them
* facilitate users to contact the uploader with the mentioned number or e-mail

Non-functional Requirements - How should it do it?

The system should be…

* available 24/7
* passwords are securely stored in a database
* any action in the system should not take more than 3 seconds to complete

**Use Cases :**

Title : TakeItFree

Primary Actor : User

Success Scenario : The user registers with the help of the registration field. If the password doesn’t meet the strength requirements or passwords do not match, app sends an error. The system allows the user to choose whether to upload or book an item. Uploader adds information about the item along with photos. If the photo is not uploaded, the user will not be able to continue uploading. Booker adds needed item from the listed item by clicking the button “Add to Cart”. The user puts the desired item in the cart, confirms to book it and contacts the uploader with the mentioned number or e-mail.

Identifying the objects :

* User Booker
* Registration field System
* Item Uploader Cart
* Information Number
* Photos E-mail

Identifying classes:

* User
* Item
* Cart
* CartItem

**Association Relationships between Classes :**

1. User and Item: Many-to-One Relationship (one item is associated with one user as the “Uploader”, and also another “User” who booked the item)
2. User and Cart: One-to-One Relationship (one user can have just only one cart, and each cart is associated with one user)
3. Cart and CartItem: One-to-Many Relationship(each cart can have multiple CartItem instances, but each cartitem belongs to one cart)
4. Item and CartItem: One-to-Many Relationship(each item can have multiple CartItem instances, but each CartItem belongs to one Item)
5. Cart and Item : Many-to-Many Relationship ( each cart contain multiple items, and each item can be part of multiple carts)

